National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

NL-339 (r. 88/04)

Canada

THE UNIVERSITY OF ALBERTA

SOURCE CURRENT DENSITY MAPPING OF THE EEG

BY

CHARLES ROBERT ARTHUR

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND
RESEARCH IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1988

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-42883-X

THE UNIVERSITY OF ALBERTA
RELEASE FORM

Name of Author: CHARLES ROBERT ARTHUR

Title of thesis: SOURCE CURRENT DENSITY MAPPING OF THE EEG

Degree: MASTER OF SCIENCE

Year this degree granted: SPRING, 1988

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

C. R. Arthur

10 Gloucester Avenue
Sherwood Park, Alberta
T8A 3B2

Date: April 20/88

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled:

SOURCE CURRENT DENSITY MAPPING OF THE EEG,
submitted by:
Charles Robert Arthur

in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE

ZJ Koles
(Supervisor)

G. D. Cormack

M. B.

Wayne A Davis

Date: 12 apr 88

To my wife Dianne
who fills my life with peace and joy

ABSTRACT

The human electroencephalogram (EEG) has a long history of clinical application, for correlating various pathological processes with brain electrical activity. Localization of the physical sources of brain electrical activity remains a more elusive goal.

Traditional methods of EEG interpretation are problematic in that they are largely visual, subject to inconsistencies resulting from differing experience levels between clinicians. There is an additional and largely ignored problem associated with the choice of an electrical reference in certain electrode / amplifier configurations. Modern techniques are exploiting the benefits of recent improvements in the cost / performance ratio of digital computers, by automating much of the interpretation process. This is significant because of the large volumes of information generated by the EEG.

This thesis focuses on a technique called topographic mapping. Here, the computer is used to generate color graphic representations of the topology of brain electrical activity as measured on the scalp. Bi-cubic splines are investigated, to provide smoother graphic representations. The Laplacian is investigated in conjunction with bi-cubic spline interpolation, to provide topological representations of scalp currents rather than voltages. Standard digital filtering techniques are applied to reduce the EEG into its constituent rhythmicities. The scalp currents obtained through the application of the Laplacian to scalp potentials flow perpendicular to the scalp and are independent of the choice of electrical reference in the unipolar montage. An additional benefit of these 'radial' current representations is that based on a dipole model of cortical generators, they serve to better localize these generators.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. J.R. Thompson, Associate Dean in the Faculty of Graduate Studies and Research and Drs. A.M. Robinson and R.P.W. Lawson, who served as Associate Chairmen for Graduate Studies in the Department of Electrical Engineering during my program. With their support, this work was made possible.

I extend special thanks to my supervisor Dr. Zoly Koles, who endured this manuscript twice. You encouraged me to dig deeper and I'll always remember your words "it's nice to exert the will over the flesh once in a while". Thank you also, to my committee members, Dr. G. Cormack and Dr. N. Durdle of the Department of Electrical Engineering and Dr. W. Davis of the Department of Computing Science, for the time you put into this thesis, and for your helpful suggestions.

To my wife Dianne, Dr. J. Lind, Raman Paranjape, Mike Lazar and Anthony Soong, who donated their valuable time proof reading various parts of this manuscript, thank you very much. To Bob Morse, Ion Buicliu and Bob Heath, three men without whom so much would never get done and whose praises should always be sung, thank you for your stimulating support. To my friend Elias Haska, a man of great courage and determination, "honor to your work comrade"! To Karim Damji, I am honored to be able to call you my friend, and wish you every success. To Dr. Kasmia Abdelhakim, a truly amazing fellow, I have been privileged to work with you and wish you good fortune and good health. Finally, to all my friends and colleagues in the Department of Applied Sciences in Medicine, you are a blessing to me and I thank God for you all. I will remember our times together as some of the most exciting in my life. Shalom!

TABLE OF CONTENTS

Chapter 1 INTRODUCTION 1
- 1.1 Describing the EEG 3
- 1.2 Some relevant developments in the topology of the EEG 6
- 1.3 Issues affecting the clinical utility of the EEG 9

Chapter 2 THE THEORETICAL BASIS 13
- 2.1 Interpolation 13
  - 2.1.1 Testing the interpolation 13
  - 2.1.2 Methods of spatial interpolation 14
    - 2.1.2.1 Low Pass filtering 14
    - 2.1.2.2 Piecewise polynomial approximation 15
    - 2.1.2.3 Bi-cubic spline interpolation 21
- 2.2 Digital Filtering techniques 26
  - 2.2.1 IIR filter design 26
  - 2.2.2 FIR filter design 27
    - 2.2.2.1 FIR filter details 28
- 2.3 The Electroencephalogram 35
  - 2.3.1 Recording configurations 36
    - 2.3.1.1 Bipolar recordings 37
    - 2.3.1.2 Unipolar recordings 38
  - 2.3.2 The reference electrode 39
    - 2.3.2.1 The Laplacian 40

Chapter 3 SYSTEM DEVELOPMENT AND PERFORMANCE 41
- 3.1 System hardware 41
  - 3.1.1 Data Acquisition 41
  - 3.1.2 Mapping hardware 44
    - 3.1.2.1 VME 10 microcomputer 44

3.2 System software 48
  3.2.1 WRTAPE 49
    3.2.1.1 Creating the maps 49
    3.2.1.2 Storing Topographic maps 50
    3.2.1.3 Evaluating topographic maps 51
    3.2.1.4 Scaling the topographic maps 52
    3.2.1.5 Filtering the data 53
    3.2.1.6 Program flow 53
  3.2.2 RDTAPE 59
    3.2.2.1 Program flow 59
3.3 Interpolation methods 63
  3.3.1 Frequency response characteristics 64
  3.3.2 One dimensional spline behavior 68
  3.3.2 Two dimensional spline behavior 70

Chapter 4 PRESENTATION OF TEST RESULTS 74
4.1 1-D spline test results 74
4.2 2-D spline test results 82

Chapter 5 DISCUSSION AND CONCLUSIONS 90
5.1 Electrical reference 90
5.2 Spatial Sampling 91
5.3 Topographic presentation 92
  5.3.1 Signal conditioning 92
  5.3.2 Bi-cubic spline interpolation 93
    5.3.2.1 Cubic spline evaluation 93
    5.3.2.2 Bi-cubic spline evaluation 94
  5.3.3 Displaying topographic maps 95
5.4 Future prospects 96
  5.4.1 System complexity 97
  5.4.2 System performance 97
  5.4.3 Parallelism 99
    5.4.3.1 Mapping system parallelism 100
    5.4.3.2 Software parallelism 101
5.5 Summing up 103

REFERENCES 106

APPENDIX A 111

APPENDIX B 140

APPENDIX C 155

LIST OF FIGURES

Figure 1.1 The International 10 - 20 System of electrode placement 4
Figure 1.2 Expanded electrode array used in this report 5
Figure 1.3 Normal EEG showing alpha rhythm 7
Figure 1.4 Abnormal EEG showing sharp transient behavior characteristic of epilepsy 8
Figure 1.5 Color scale used for topographic mapping 12
Figure 2.1 Expanded electrode array used in this report 14
Figure 2.2 Continuity conditions for the Bi-cubic spline 23
Figure 2.3 Conditions for uniqueness of the bi-cubic spline 24
Figure 2.4 Necessary boundary conditions to uniquely solve the bi-cubic spline 25
Figure 2.5 Frequency response for an ideal low-pass digital filter 29
Figure 2.6 Low pass filter 0 - 15 Hz Rectangular windowed to 20 terms 31
Figure 2.7 Low pass filter 0 - 15 Hz Rectangular windowed to 50 terms 32
Figure 2.8 Low pass filter 0 - 15 Hz Hamming windowed to 50 terms 34
Figure 2.9 Low pass filter 0 - 15 Hz Hanning windowed to 50 terms 35
Figure 2.10 The International 10 - 20 Systemof electrode placement 36
Figure 2.11 Recording configurations 37
Figure 3.1 A typical recording session 42
Figure 3.2 The Neurosciences machine 43
Figure 3.3 The digitizing system 43
Figure 3.4 Front view of the VME 10 development system 45
Figure 3.5 Rear view of the VME 10 development system showing VMEbus modules 46
Figure 3.6 Mapping array 50
Figure 3.7 Display configuration 51
Figure 3.8 Frequency response plots of Delta band, Theta band and Alpha band filters 54
Figure 3.9 WRTAPE flowchart 56
Figure 3.10 RDTAPE flowchart 62
Figure 3.11 Point spread function (chateau function) for linear interpolation 66

Figure 3.12 (a) Point spread function for cubic spline interpolation using the 'free boundary condition 66
Figure 3.12 (b) Point spread function for cubic spline interpolation using the 'clamped'boundary condition 67
Figure 3.13 Comparison of the frequency response for linear and cubic spline interpolation, with and ideal interpolator 67
Figure 3.14 (a) Spline interpolation of a sampled actual function $f(x) = \cos 1.6x$ 69
Figure 3.14 (b) Spline interpolation of a sampled actual function $f(x) = \cos(1.6x+73^\circ)$ 69
Figure 3.15 Bicubic spline test patterns 71
Figure 3.16 Corner value test patterns 72
Figure 4.1 (a) Spline of an actual function $f(x) = \cos x$ 75
Figure 4.1 (b) Spline of an actual function $f(x) = \cos(x+45^\circ)$ 76
Figure 4.1 (c) Spline of an actual function $f(x) = \cos 1.5x$ 76
Figure 4.1 (d) Spline of an actual function $f(x) = \cos(1.5x+45^\circ)$ 77
Figure 4.1 (e) Spline of an actual function $f(x) = \cos 1.8x$ 77
Figure 4.1 (f) Spline of an actual function $f(x) = \cos(1.8x+45^\circ)$ 78
Figure 4.1 (g) Spline of an actual function $f(x) = \cos 2x$ 78
Figure 4.1 (h) Spline of an actual function $f(x) = \cos(2+45^\circ)$ 79
Figure 4.1 (i) Spline of an actual function $f(x) = \cos 4x$ 79
Figure 4.1 (j) Spline of an actual function $f(x) = \cos(4x+45^\circ)$ 80
Figure 4.1(k) Spline of an actual function $f(x) = \cos 6x$ 80
Figure 4.1(l) Spline of an actual function $f(x) = \cos(6x+45^\circ)$ 81
Figure 4.1(m) Spline of an actual function $f(x) = \cos x + 1.2\cos(1.2x+10^\circ) + 0.8\cos(1.4x+20^\circ)$ 81
Figure 4.1 (n) Spline of an actual function $f(x) = \cos x + 1.2\cos(1.2x+10^\circ) + 0.8\cos(1.4x+20^\circ) + \cos(1.8x+30^\circ) + 1.2\cos(2x+40^\circ)$ 82
Figure 4.2 (a) Interpolated voltage map for test pattern 3.15 (a) (clamped and free boundary conditions) 83
Figure 4.2 (b) Interpolated voltage map for test pattern 3.15(b) (clamped and free boundary conditions) 83
Figure 4.2 (c) Interpolated voltage map for test pattern 3.15 (c) (clamped and free boundary conditions) 84

Figure 4.3 Interpolated voltage map for test pattern 3.16 (a)
(clamped and free boundary conditions) 85
Figure 4.4 (a) Interpolated voltage map for test pattern 3.16 (b)
Averaged and zeroed corners (clamped boundary condition) 86
Figure 4.4 (b) Interpolated voltage map for test pattern 3.16 (b)
Averaged and zeroed corners (free boundary condition) 86
Figure 4.5 (a) Current maps of actual data (Alpha rhythm)
Averaged corners (clamped boundary condition) 87
Figure 4.5 (b) Current maps of actual data (Alpha rhythm)
Zeroed corners (clamped boundary condition) 88
Figure 4.5 (c) Current maps of actual data (Alpha rhythm)
Averaged corners (free boundary condition) 88
Figure 4.5 (d) Current maps of actual data (Alpha rhythm)
Zeroed corners (free boundary condition) 89
Figure 5.1 A possible multi-processor configuration
for high speed topographic mapping 103
Figure C.1 WRTAPE histogram 155
Figure C.2 RDTAPE header file information 156
Figure C.3 RDTAPE channel numbering pattern 156
Figure C.4 Typical RDTAPE raw EEG data display 157
Figure C.5 Larger format for cartooning display 158
Figure C.6 18 contiguous map page type display 158

Chapter 1

INTRODUCTION

In 1929, the first in a series of 14 papers entitled "On the electroencephalogram of man" was published by Hans Berger, a relatively unknown German professor of psychiatry at Jena. At this time, the study of the central nervous system physiology was in its infancy and was little more than functional neuroanatomy. Pierre Gloor [24] describes the study of neurophysiology at the time,

> "Only Sherrington in his work on the spinal cord had begun to come to grips with the physiological principles of the central nervous system function. These principles, however, were still derived from observations of events taking place outside the central nervous system, such as the carefully measured pull or relaxation of muscles in response to well defined stimuli."

The scientific community took little notice of Berger, who had "leapfrogged across all the intermediate stages ... of the Sherringtonian approach .. and addressed himself directly to the electrophysiological investigation of the most complex functions of the human brain". Berger himself was not particularly well schooled or interested in electrophysiology, but following his own particular interests in psychophysiology [24],

> "he believed that ... he had found the key which would unlock the secret of man's nature as a psychophysical being. He believed that in the EEG he was able to discern some well defined and measurable physical properties which represented true expressions of mental processes.".

Berger continued to publish his findings, and by 1934 noted neurophysiologists Adrian and Matthews, because of their admitted skepticism, repeated Berger's experiments. As a result, they verified Berger's findings. Berger's work now began to be accepted outside Germany and the era of the EEG had begun. As the 60th anniversary of the founding of the EEG approaches and although it has benefitted much from the advance of technology, the manner in which the EEG is presented to its practitioners has changed little, and it is given only a qualitative visual interpretation.

Shipton [38] observed in 1971 that "most progress ... has been in medicine: empirical correlations between disease processes and electrical activity of the brain have firmly established the technique in the clinic". Shipton points out that,

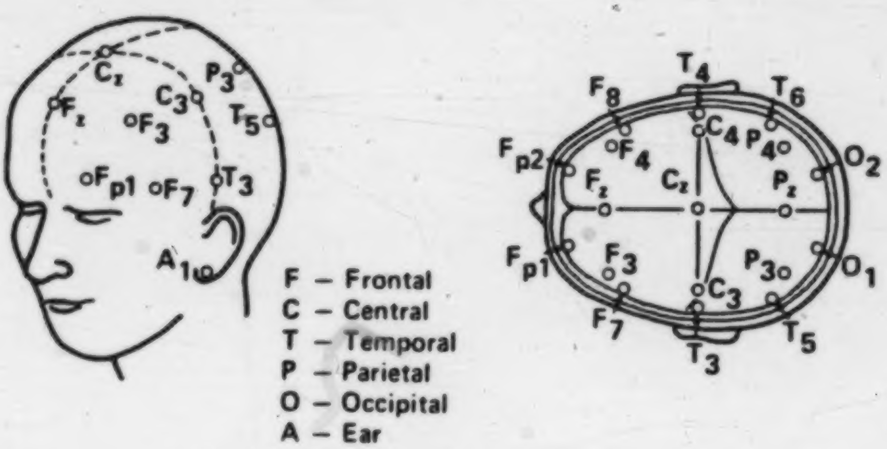
> "although the alpha rhythm has been studied for forty years and whilst it has been the subject of innumerable papers, we are not even close to a generally held hypothesis as to its psychophysiological significance".

Although this situation has been alleviated somewhat [13], the manner in which the EEG is presented and interpreted has changed little over the years, in spite of the strength of signal processing techniques already available by the mid 1930's, which did in fact see some application. Von Neumann and the digital computer were not to come along until 1945, the ubiquitous Fast Fourier Transform (FFT) did not appear until 1966 [5] and the proliferation of powerful, portable and low cost digital computing technology did not even begin to appear until 1980. In addition, Shipton [38] points to a "mathematical naïveté of many clinicians active during the 1950s" and Nunez [35] points to a need to bridge "the communication gap between the electroencephalographer and physicist". Nunez illustrates how electrophysics can bring much knowledge to bear upon the qualitative understanding of the physiological basis for the EEG and thus significantly improve its interpretation. Fortunately, the above mentioned advances have occurred, along with significant advances in understanding the electro-chemical aspects of neurophysiology and detailed microscopic and macroscopic understanding of the organization of the brain and central nervous system. Thus, modern electroencephalography seems to be on the verge of some significant breakthroughs for both clinical utility and increasing man's knowledge of himself from a psychophysical point of view.

In the above and ensuing discussion, the author does not wish to imply an intimate knowledge of the various fields of neurology, electrophysiology, neurophysiology or neuroanatomy. The purpose of this historical review is to review the expert knowledge reported by others, so that the contributions this work makes to contemporary electroencephalography might be properly classified.

1.1 Describing the EEG

It is useful to discuss some common terminology frequently used in the field of modern electroencephalography. To begin with, observed brain electrical activity can be divided into 2 categories: that related to the brain's response to external stimuli is termed exongenous functioning, while that which is related to internal processing is termed endogenous functioning. Endogenous functioning accounts for about 90% of the background activity, or what Berger termed the "passive EEG". Brain electrical activity as measured on the scalp is between 5 and 200μV. That endogenous functioning accounts for 90% of the passive EEG is due to the knowledge that generators of this activity lie in the grey matter, or cerebral cortex, which is the outer 2-3 millimeter thick layer of the brain and thus these generators are more proximal to the scalp electrodes. The passive EEG is recorded by an array of electrodes placed on the scalp and connected in either a bi-polar or unipolar montage discussed in detail in chapter 2. Figure 1.1 illustrates the International 10/20 System, while figure 1.2 illustrates the expanded electrode array used in the work reported here. Generators of exogenous functioning on the other hand, are deep within the brain's white matter, where the relay stations [35] associated with the brain's early response to external stimuli are located. Thus, these generators are farther from the scalp electrodes and their activity is buried in the 'noise' of endogenous functioning. Computer advances [2] enabled electroencephalographers to apply the technique of coherent averaging. Here, the EEG is recorded in coordination with the repeated application of various sensory stimuli. Coherent averaging effectively increases the signal to noise ratio of exogenous / endogenous functioning allowing the observer to observe repeatable electrical patterns which have been termed "sensory evoked responses" or "Evoked Potentials" (EP). This thesis has addressed itself to the passive EEG, although the topographic mapping techniques to be described can easily be applied to EP's as has been done by Duffy [10].


Cz
P3
Fz
C3
T5
F3
Fp1
F7
T3
A1
F – Frontal
C – Central
T – Temporal
P – Parietal
O – Occipital
A – Ear
T4
F8
T6
Fp2
F4
C4
P4
O2
Fz
Cz
Pz
Fp1
F3
P3
O1
C3
F7
T5
T3

The International 10 - 20 System
of electrode placement
Figure 1.1

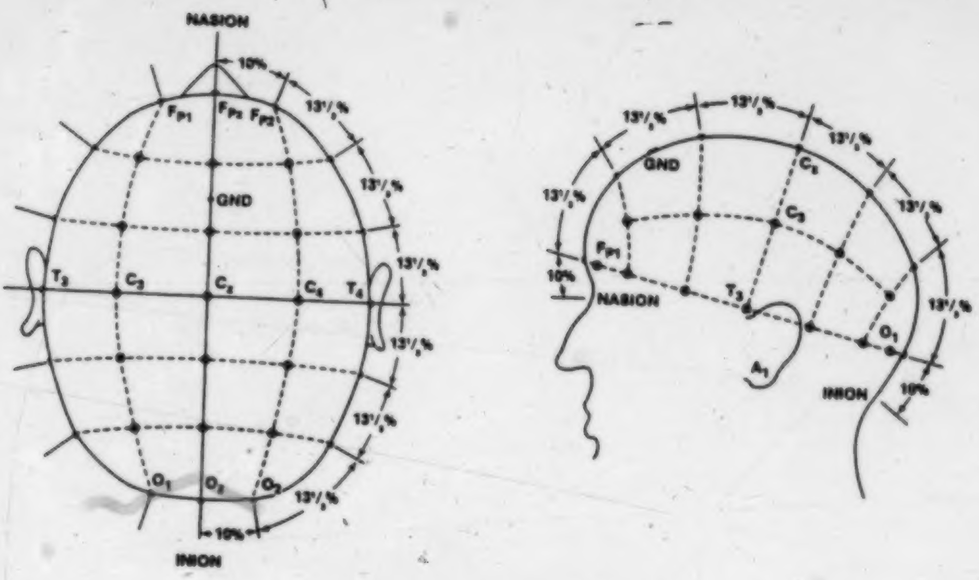
The spectral characteristics of the passive EEG have over the years been classified into the following areas:

Alpha rhythm

This is the quasi sinusoidal rhythm, first observed by Berger, characteristic of a relaxed but alert individual and is enhanced when the eyes are closed. The alpha rhythm is generally present over the entire scalp but predominates at the back of the head over the occipital region of the brain: it is most characteristic of the EEG of humans but to a lesser extent appears in the EEG of monkeys and dogs. The alpha rhythm lies in the frequency band 8 to 13 Hz., is initially not present in infants but increases in strength through childhood to stabilize at about 10 Hz. at around 12 years of age. In middle to old age, the alpha rhythm begins to decrease in frequency and amplitude.

Beta rhythm

Berger was again the first to observe the beta rhythm, which is also a quasi sinusoidal rhythm composed of frequencies above 13 Hz. Beta occurs when a subject opens the eyes, is externally stimulated, or begins to perform some cognitive task. Sometimes the beta frequency band is divided into beta$_1$ (13-20 Hz) and beta$_2$ (>20 Hz).



Expanded electrode array used
in this report
Figure 1.2

Theta band

As a subject becomes drowsy and enter the early stages of sleep, the alpha rhythm begins to slow, lose its sinusoidal character and become less synchronous. The theta frequency band is nominally between 4 and 8 Hz.

Delta rhythm

These are rhythmic waves, though less sinusoidal, present during deep sleep in normal people of all ages and are the primary waves present in the recordings of normal infants. Delta waves are almost always pathological in the waking records of adults.

As sleep deepens, delta waves become larger in amplitude, slower and more synchronized, until rapid eye movement (REM) occurs. REM, characteristic of the dream stage, when the sleeper makes rapid horizontal eye movements as if scanning a real scene, is the most difficult stage of sleep from which to awaken a subject. Yet, during REM, the EEG is more like that of an awake individual, with alpha activity present.

Berger [24] "was the first to describe the presence of slow waves in the EEG associated with many kinds of cerebral lesions. He interpreted them as slow alpha waves and was unwilling to accept the term delta waves".

To quote a recent grant application by Dr. Koles / Dr. Mclean "clinical electroencephalography is probably most often used for seizure treatment". Berger [24] "made many interesting observations on epilepsy". In a surgical environment, the EEG can be used to ascertain the depth of anaesthesia. The EEG is also typically used to study behavioral disorders and mental retardation, particularly in children. The ability to diagnose pathologies was alluded to above: such pathologies include tumor location, indicating brain damage, stroke and death. Further applications of the EEG include the study of sleep, the effects of drugs and cognition.

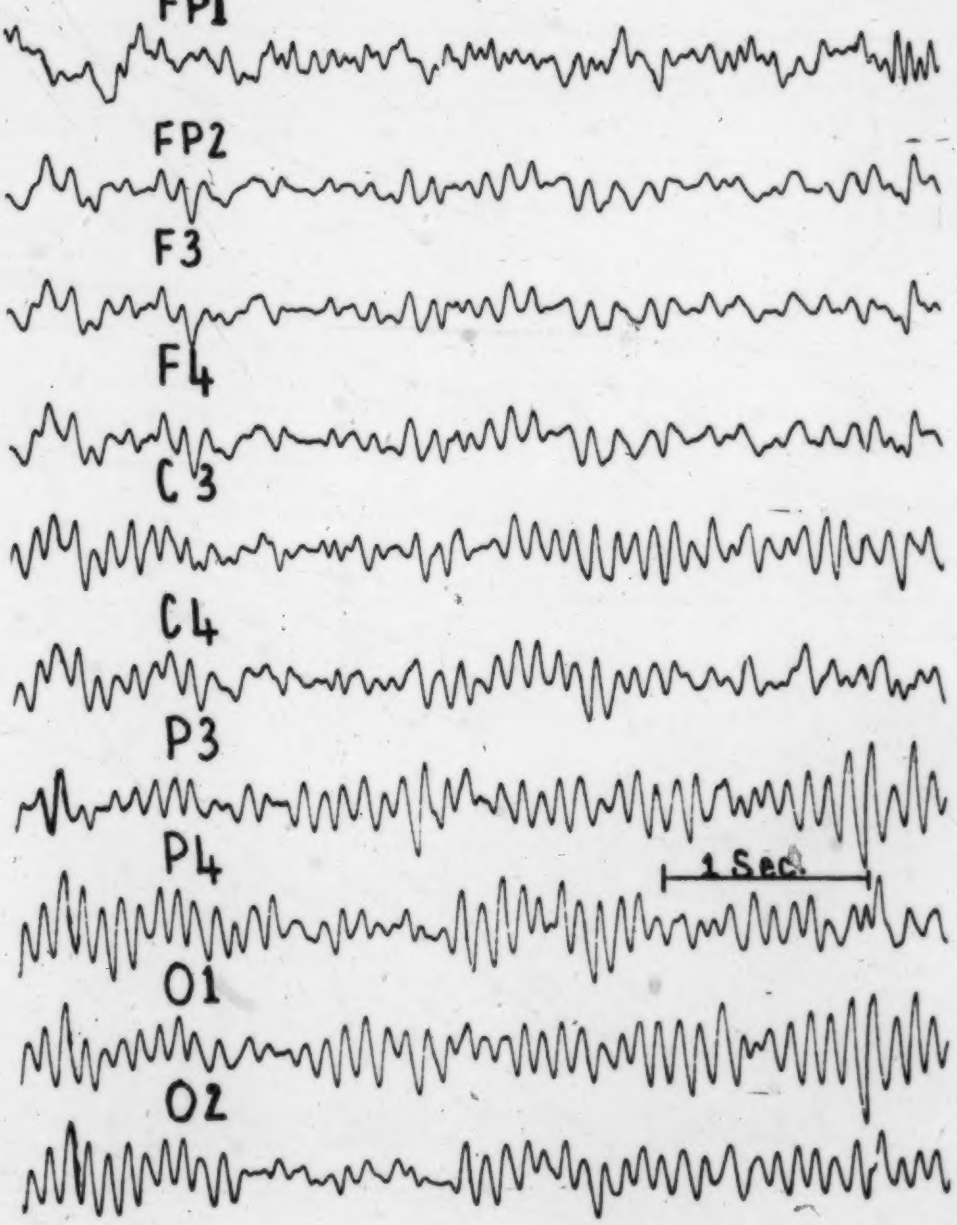
To classify the EEG in relation to the above pathologies, 3 major classes of abnormality are considered:

- asymmetries between hemispheres
- slow rhythms
- very sharp waves or spikes.

Sharp waves and spikes were correctly associated by Berger [24] with hyper excitability or irritable state of the cortex and a predisposition toward seizures. During a full epileptic seizure attack, spikes become repetitive and synchronized over the whole surface of the brain. The pathologies associated with slow waves were discussed above. Hemispherical asymmetries are typically associated with tumors and functional lesions.

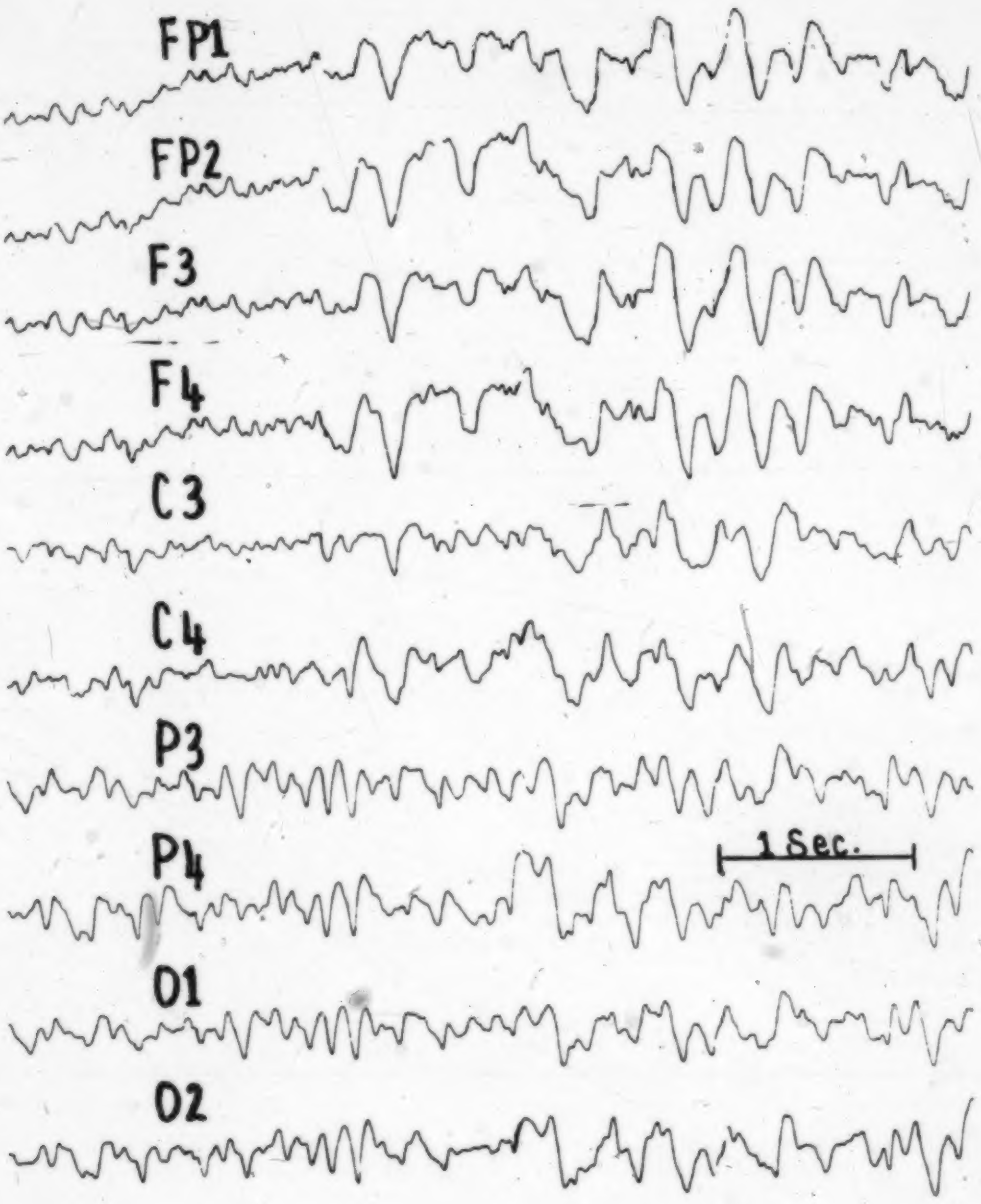
1.2 Some relevant developments in the topology of the EEG

From the beginning, Berger and those who pursued the study of the EEG have been interested in its topology. Berger was limited by never having had the benefit of a multi-channel oscillographic recording machine, which was developed much later. Despite many modern advances in the technology supporting the EEG, it is given only qualitative visual interpretation in which case the utility of the EEG is dependant upon the skill and experience of the clinician. In developing these skills, electroencephalographers have had to learn to do visual spectral analysis and mentally assimilate the spatial properties of the polygraphic recording of the EEG. In addition, they have had to become familiar with spikes and transient responses indicative of various pathologies or muscle movement. Although the modern electroencephalographer has become very adept at these tasks, it remains a tremendously demanding and stressful task which few experienced interpreters are able to perform. The reason for this difficulty, is that the EEG generates large volumes of multi-dimensional data. If this multi-dimensionality can be consistently decomposed by machine, the clinical utility of the EEG can be significantly improved. Figures 1.3 and 1.4 illustrate typical polygraphic recordings of normal and abnormal EEG, providing some insight into the difficulties facing the clinical electroencephalographer.


FP1
FP2
F3
F4
C3
C4
P3
P4
1 Sec.
O1
O2

Normal EEG showing strong alpha rhythm

Figure 1.3


FP1
FP2
F3
F4
C3
C4
P3
P4
1 Sec.
O1
O2

Abnormal EEG characterized by slowing activity in the delta band

Figure 1.4

Subsequent to the development of the FFT [5], Bickford et al [2,38] in 1972 addressed the topology problem by developing the compressed spectral array (CSA). In this technique, FFT's are computed for contiguous epochs of the EEG from each electrode, and displayed in a topological fashion inside a descriptive outline of the head from a coronal view. This display method is common in computerized EEG systems currently available.

Other researchers [2,8,10,14-20,30,31] approached the topology problem by considering the unipolar montage electrode array as a spatial sampling of the electrical potential on the scalp and have thus interpolated between electrode positions. Perhaps the most successful of these 'topographic mapping' proponents to emerge during the mid 1970's, at least from a clinical perspective, is Dr. Frank Duffy [10,8] who developed the BEAM (brain electrical activity mapping) system which he has used in a number of clinical studies of the passive EEG and evoked potentials of tumor patients, dyslexics and schizophrenics. Still another researcher to study topographic mapping is Gevins [13,21], although the bulk of his prolific activity has been to study the topological and statistical properties of the cognitive EEG [14-20].

In addition to the development of topographic mapping, there is a considerable body of work which has developed concerning the computer recognition of spike waveforms as well as the recognition and rejection of artifact. This activity however, is considered outside the scope of this work, which addresses itself to improving the utility of the EEG by decomposing it into it's rhythmic components and using interpolation to produce topographic maps which readily illustrate the spatial EEG.

1.3 Issues affecting the clinical utility of the EEG

It is possible to establish 4 major issues which, if resolved would contribute significantly to improving the clinical utility of the EEG.

1) Presentation of the EEG
2) Sampling the spatial EEG
3) Choosing an electrical reference for the EEG
4) Characterization of EEG patterning

This thesis addresses itself to the resolution of the first 3 of the above, which will now be considered in turn.

Presentation of the EEG

It was mentioned above that presentation of the EEG could be significantly improved by decomposing it into its rhythmic components and using interpolation to produce topographic maps which readily illustrate the spatial EEG. Furthermore, it is suggested that the linear interpolation methods used to date are inadequate from the point of view that the cerebral electric activity as measured on the scalp can reasonably be considered as being a smooth continuous surface. Thus, a higher order interpolation method known as bi-cubic splines will be used. Perrin, Bertrand and Pernier [36] have already reported the use of natural spline interpolation in simulation studies.

Sampling the spatial EEG

It is a well understood concept in digital signal processing, that if interpolation is to be valid, sampling density should be twice that of the highest constituent frequencies in the underlying function. To quote Dr. Koles / Dr. Mclean again, "this theorem can also be expressed in terms of a point-spread function which in essence is a measure of the spatial extent to which the presence of a point generator can be measured in a volume conductor". Gevins [22] has suggested, that in the brain, the point-spread function is about 1 cm. for a source on the cortex. Thus, in some situations at least, more than 200 recording electrodes would be necessary to adequately sample the spatial EEG. It is an open question, however, whether sources in the brain are point sources or whether they are sometimes extensive sheets [35] which would indicate that much less than 200 electrodes would be required. Duffy [8,10] uses the more normal 16 to 20 electrodes while Lehmann [30] uses 48 but does not interpolate. The work of this report uses a 31 electrode configuration, and employs bi-cubic spline interpolation.

Choosing an electrical reference for the EEG.

This is perhaps the most central issue, which this thesis will address. The reference problem is discussed in detail in chapter 2, but in summary, it is impossible to find an inactive location on the body to place a reference electrode. Placing one on the head makes it impossible to distinguish the reference electrode from all other electrodes in the unipolar montage. The single ear reference has been criticized because it is near a hole in the skull, a location of reduced resistance, and thus quite active. The linked ears reference is also criticized because it short circuits any potential difference which may otherwise exist between the ears. The average reference, which has become popular recently, utilizes as a reference the average potential of all other locations in the electrode array. This is simple to compute if the EEG data is available in digital form, but it is still not an inactive reference. Thus, caution must be used to interpret recordings so obtained, since the activity seen at any particular location depends on activity present at all other electrodes, which is in turn influenced by the location of these other electrodes.

This thesis utilizes the reference free solution proposed by Katznelson [29], otherwise known as the source derivation method, which is based upon determining the spatial divergence of the tangential current density field in the scalp. It will be shown in chapter 2, that this spatial divergence can be determined by applying the Laplacian operator to the scalp potential. An added benefit of using the source deviation method is that it produces topographic maps which portray 'radial current' flow in the scalp which in theory will better localize generators of cortical activity [29]. Perrin et al. [36] have also use the source deviation method in their dipole model simulation studies. Because the bi-cubic spline interpolation method is used in this work, the Laplacian can be applied analytically to the interpolating polynomials, representing a significant improvement over numerical approximations to the Laplacian, necessary with linear interpolation.

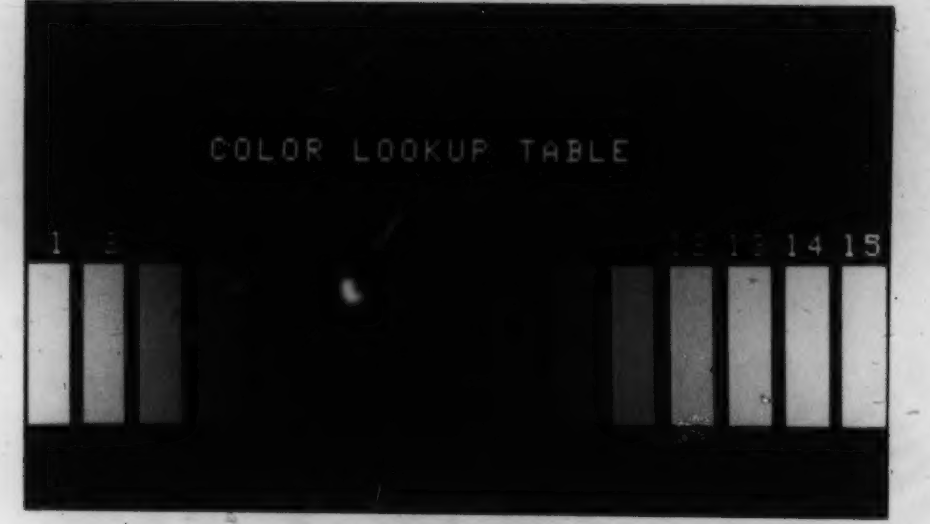
As a final consideration, the color encoding of scaled topographic maps was considered. A psuedo color scale was developed which would aid in the user's perception of the electrical variations represented by the voltage or source current maps produced. The color scale developed uses a pale green of relatively low intensity to distinguish the zero level from the black background. Increasing intensity was employed in all other colors used, to portray increasing electrical intensity in either the positive or negative directions. In the negative direction,blue was used to portray the lowest level, increasing in intensity and changing color to become more violet as electrical level increased. Similarly, a pale green to bright white scale was employed in the positive direction. The color scale employed was placed in the 16 color lookup table of the color graphics processor and is portrayed in the color photograph of figure 1.5

THE QUALITY OF THIS MICROFICHE IS HEAVILY DEPENDENT UPON THE QUALITY OF THE THESIS SUBMITTED FOR MICROFILMING.

UNFORTUNATELY THE COLOURED ILLUSTRATIONS OF THIS THESIS CAN ONLY YIELD DIFFERENT TONES OF GREY.

LA QUALITE DE CETTE MICROFICHE DEPEND GRANDEMENT DE LA QUALITE DE LA THESE SOUMISE AU MICROFILMAGE.

MALHEUREUSEMENT, LES DIFFERENTES ILLUSTRATIONS EN COULEURS DE CETTE THESE NE PEUVENT DONNER QUE DES TEINTES DE GRIS.


COLOR LOOKUP TABLE
1
13
14
15

Color scale used for topographic mapping

Figure 1.5

Chapter 2
THE THEORETICAL BASIS

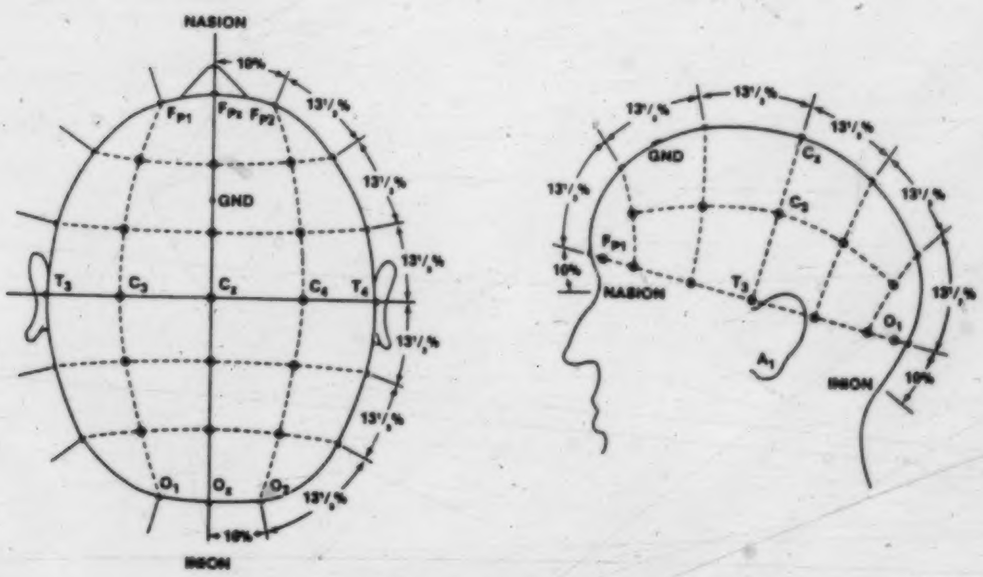
2.1 Interpolation

To do topographic mapping, the electrode montage is viewed as a two-dimensional spatial sampling of the scalp electric potential surface. Sampling theory states that when a band-limited signal is sampled, the sample set produced is not unique to the sampled signal, but that there are an infinite number of frequency bands identical in shape to the baseband, but centered at integer multiples of the sampling frequency, which will produce the same set of samples. This leads to the Nyquist sampling criterion, which says that if a band-limited signal is sampled at a minimum of twice the highest frequency in the baseband, an ideal low-pass filtering operation will perfectly reconstruct the baseband signal. In this sense, the impulse response of the ideal low-pass filter is sometimes referred to as the ideal interpolating function. Interpolation then, can also be viewed as a low-pass filtering operation, and in any attempt at image reconstruction, it must be established that the image was adequately sampled, if a valid reconstruction of the underlying image is to be produced.

2.1.1 Testing the interpolation

The method of interpolation used in this project was the bicubic spline and will be discussed in detail later. Preliminary testing of this interpolation method focused on the ability of the interpolant to make accurate and precise estimates of inter electrode potentials. Test data were obtained from one normal individual. In the test, 25 electrodes were located to achieve a near rectangular sampling of the scalp; 5 additional electrodes were placed in inter electrode locations. Recordings were then taken and inter electrode values predicted by interpolation compared with those measured. The test showed that the bi-cubic spline gave accurate and precise estimates when the temporal frequency was limited to 13 Hz. and when scalp electrode potential was above 5μV.

It is well known, that the accuracy of the interpolation process depends directly upon the adequacy with which the original signal was sampled as discussed above. With the spatial EEG, little detailed information is available on the full extent of the baseband for these signals. Gevins [22] has suggested that as many as 200 electrodes may be necessary to adequately sample the spatial EEG in some cases. In the absence of this detailed information on the spatial EEG, the experiment noted above shows that the 25 electrode sampling is adequate in at least some cases. For this work, the sampling density was increased to a 31 electrode arrangement illustrated in figure 2.1 and discussed further in chapter 3 . It is reasonable to expect then, some improvement in the accuracy of the interpolations performed and that a basis has been established to further develop topographic mapping systems for the EEG at these spatial sampling densities.



Expanded electrode array used
in this report
Figure 2.1

2.1.2 Methods of spatial interpolation

2.1.2.1 Low Pass filtering

To start this chapter, the point was made that interpolation can be considered as a form of low-pass filtering . Thus, to interpolate a sampled two dimensional image, a 2-D low-pass finite impulse response (FIR) filter can be constructed, appropriate to the amount of interpolation required. Such a 2-D FIR filter could be applied using either linear convolution in the spatial domain, or multiplication in the frequency domain [7].

In this work, the electrode montage used was viewed as a 5 by 7 point square sampling of the scalp potential surface. In the low-pass filtering process, some assumption must be made about the image values outside the boundaries, or region of support of the image. Typically, the image is considered to be zero outside the image's region of support. A bi-product of the filtering process is 'ringing' that results from the assumed jump discontinuity at the boundary. The size of this area of 'ringing' is directly dependant on the size of the filter used.

2.1.2.2 Piecewise polynomial approximation

Single polynomials of high degree can be quite oscillatory in nature and fluctuations over a small portion of the interval can induce large fluctuations over the entire range [3] . These characteristics often compromise their ability to perform in an actual physical situation such as that presented by the EEG. Alternatively, the interval can be divided into subintervals and a different approximating polynomial constructed over each subinterval. This allows the approximating function the flexibility to adjust to local variations in a subinterval without having a large effect on approximations outside the subinterval. Such a technique is referred to as piecewise polynomial approximation, the simplest being linear interpolation.

Smoothness over the interpolating interval is important, so that higher order polynomials with multiple differentiability are of interest. There are many ways to construct higher order polynomials, but with the EEG, nothing is known about the continuity of the scalp potential surface at the measuring points other than the potential values themselves. Also, computational complexity is a consideration, and so the use of simple cubic polynomials is a logical next step.

Piecewise polynomial approximation using cubic polynomials between successive pairs of measurement points, or nodes, is called cubic spline interpolation. The cubic spline has the flexibility of double differentiability to establish continuity between subintervals. Consider then, the following definition [3] for the one dimensional case.

> Given a function f defined on an interval [a ,b] and a set of numbers $a = x_1 < x_2 < \ldots < x_n = b$, a cubic spline interpolant, S, for f is a function that satisfies the following conditions:
>
> a) S is a cubic polynomial, denoted S_k, on the subinterval $[x_k, x_{k+1}]$ for each k = 1,2,3, ... ,n-1;
> b) $S(x_k) = f(x_k)$ for each k = 1,2,3, ... ,n;

c) $S_{k+1}(x_{k+1}) = S_k(x_{k+1})$ for each $k = 1,2,3, \ldots ,n-2$;
d) $S'_{k+1}(x_{k+1}) = S'_k(x_{k+1})$ for each $k = 1,2,3, \ldots ,n-2$;
e) $S''_{k+1}(x_{k+1}) = S''_k(x_{k+1})$ for each $k= 1,2,3, \ldots ,n-2$;

and in addition, any pair of the following boundary conditions must be satisfied to provide enough conditions to determine a unique spline:

$$\ddot{S}_1\,(x_1) = K1\ ,\ \ddot{S}_{n-1}\,(x_n) = K2\ ,\ \dot{S}_1\,(x_1) = K3\ ,\ \dot{S}_{n-1}\,(x_n) = K4$$

The boundary condition $S''_1(x_1) = S''_{n-1}(x_n) = 0$ is referred to as the 'free' boundary condition, and the boundary condition $S'_1(x_1) = S'_{n-1}(x_n) = 0$ is referred to as the 'clamped' boundary condition. The 'free' boundary condition is a more general assumption to make in the absence of any knowledge of the actual slope of the function being interpolated, at the boundary points. The 'clamped' boundary condition will lead to more accurate interpolations however, if accurate assumptions can be made about the slope of the unknown function f at the boundary nodes.

In the following discussion of the one dimensional spline, the following notation will be used for simplicity:

$$S_k(x_k) = y_k,\ S'_k(x_k) = y'_k, \text{ and } S''_k(x_k) = y''_k.$$

The following discussion is excerpted from Spath [39] with additional clarification. If the cubic polynomials, are chosen to be of the form:

$$S_k = A_k(x-x_k)^3 + B_k(x-x_k)^2 + C_k(x-x_k) + D_k,$$

then the 4(n-1) coefficients A_k, B_k, C_k and D_k must be chosen to satisfy the conditions of the definition above. From these conditions, it can be determined that:

$$\begin{aligned}(2.1)\quad & y_k = S_k(x_k) = D_k\\ & y_{k+1} = S_k(x_{k+1}) = A_k\,\Delta x_k^3 + B_k\Delta x_k^2 + C_k\Delta x_k + D_k\end{aligned}$$

$$\begin{aligned}(2.2)\quad & y'_k = S'_k(x_k) = C_k\\ & y'_{k+1} = S'_k(x_{k+1}) = 3A_k\Delta x_k^2 + 2B_k\Delta x_k + C_k\end{aligned}$$

$$\begin{aligned}(2.3)\quad & y''_k = S''_k(x_k) = 2B_k\\ & y''_{k+1} = S''_k(x_{k+1}) = 6A_k\Delta x_k + 2B_k\end{aligned}$$

where $\Delta x_k = x_{k+1} - x_k$

From (2.1) and (2.3), $D_k = y_k$ and $B_k = y''_k/2$. Substituting these into the second parts of (2.1) and (2.3), and solving the resulting simultaneous equations for A_k and C_k, the following relationships are obtained:

$$A_k = \frac{1}{6\Delta x_k}\,(y''_{k+1} - y''_k)$$

$$B_k = \frac{1}{2}\,y''_k$$

$$\text{(I)} \qquad C_k = \frac{\Delta y_k}{\Delta x_k} - \frac{1}{6}\,\Delta x_k\,(y''_{k+1} + 2y''_k)$$

$$D_k = y_k$$

where $\Delta y_k = y_{k+1} - y_k$

Using (2.1) and (2.2) in a similar manner, an analogous result can be obtained:

$$A_k = \frac{1}{\Delta x_k^2}\left(-2\,\frac{\Delta y_k}{\Delta x_k} + y'_k + y'_{k+1} \right)$$

$$B_k = \frac{1}{\Delta x_k}\left(3\,\frac{\Delta y_k}{\Delta x_k} - 2y'_k - y'_{k+1} \right)$$

$$\text{(II)} \qquad C_k = y'_k$$

$$D_k = y_k$$

Note, that while (I) expresses the spline coefficients in terms of the spline second derivatives at the nodes, (II) expresses them in terms of the spline first derivatives at the nodes. As yet however, nothing is known about either the spline first derivatives or second derivatives at the nodes although assumptions will be required about them at the boundary nodes.

Consider first the equations (I). If these are substituted into the pair of equations (2.2), it can be determined that, for $k = 1, \cdots, n-1$,

$$y'_k = C_k = \frac{\Delta y_k}{\Delta x_k} - \frac{1}{6}\,\Delta x_k\,(\,y''_{k+1} + 2y''_k\,) \tag{2.4}$$

and

$$\begin{aligned} y'_{k+1} &= 3A_k\Delta x_k^2 + 2B_k\Delta x_k + C_k \\ &= \frac{\Delta y_k}{\Delta x_k} + \frac{1}{6}\,\Delta x_k\,(\,2y''_{k+1} + y''_k\,)\,. \end{aligned} \tag{2.5}$$

Using (2.4) and (2.5) and the requirement for continuity in the first derivative that

$$S'_{k-1}(x_k) = S'_k(x_k) \qquad k = 2, \ldots, n-1$$

the following result can be obtained:

$$6\left(\frac{\Delta y_k}{\Delta x_k} - \frac{\Delta y_{k-1}}{\Delta x_{k-1}}\right) = y''_{k-1}\Delta x_{k-1} + 2y''_k\left(\,\Delta x_k + \Delta x_{k-1}\,\right) + y''_{k+1}\Delta x_k\ . \tag{2.6}$$

Equation (2.6) represents n-2 equations in n unknowns $y''_1, \cdots, y''_n$. If the values for y''_1 and y''_n (boundary conditions) are known or can be assumed, (2.6) becomes n-2 equations in the n-2 unknowns $y''_2, \cdots, y''_{n-1}$. In matrix form, (2.6) is tridiagonal and diagonally dominant, which guarantees a unique solution [3,39]. Represented in matrix form equation (2.6) can be expressed as

$$\begin{bmatrix} 2(\Delta x_1+\Delta x_2) & \Delta x_2 & 0 & 0 & 0 & 0 \\ \Delta x_2 & 2(\Delta x_2+\Delta x_3) & \Delta x_3 & 0 & 0 & 0 \\ 0 & \Delta x_3 & 2(\Delta x_3+\Delta x_4) & \Delta x_4 & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . \\ 0 & 0 & 0 & 0 & \Delta x_{n-2} & 2(\Delta x_{n-2}+\Delta x_{n-1}) \end{bmatrix} \begin{bmatrix} y''_2 \\ y''_3 \\ y''_4 \\ . \\ . \\ y''_{n-1} \end{bmatrix} =$$

$$\begin{bmatrix} 6\left(\frac{\Delta y_2}{\Delta x_2} - \frac{\Delta y_1}{\Delta x_1}\right) - \Delta x_1 y''_1 \\ 6\left(\frac{\Delta y_3}{\Delta x_3} - \frac{\Delta y_2}{\Delta x_2}\right) \\ 6\left(\frac{\Delta y_4}{\Delta x_4} - \frac{\Delta y_3}{\Delta x_3}\right) \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ 6\left(\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{\Delta y_{n-2}}{\Delta x_{n-2}}\right) - \Delta x_{n-1} y''_n \end{bmatrix}$$

The discussion above provides a way to determine the second derivatives of the cubic spline at the interior nodes, given some assumed knowledge of the second derivative at the boundary. However, with the form of equations (2.6), there is no need to limit the choice of boundary value assumptions to the second derivative. If instead, boundary value assumptions for first derivative are made, equations (2.4) and (2.5) can be used to determine the following two equations for y''_1 and y''_n which can be added to the system of equations (2.6).

$$\text{(2.6a)} \qquad 2\Delta x_1 y''_1 + \Delta x_1 y''_2 = 6\left(\frac{\Delta y_1}{\Delta x_1} - y'_1\right)$$

$$\text{(2.6b)} \qquad \Delta x_{n-1} y''_{n-1} + 2\Delta x_{n-1} y''_n = 6\left(y'_n - \frac{\Delta y_{n-1}}{\Delta x_{n-1}}\right)$$

This now gives a system of n equations in the n unknowns $y''_1, \cdots, y''_n$. The coefficient matrix remains tridiagonal and diagonally dominant and is thus uniquely solvable. In matrix form, the equations become:

$$\begin{bmatrix} 2\Delta x_1 & \Delta x_1 & 0 & 0 & 0 & 0 \\ \Delta x_1 & 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & 0 & 0 & 0 \\ 0 & \Delta x_2 & 2(\Delta x_2 + \Delta x_3) & \Delta x_3 & 0 & 0 \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & \Delta x_{n-2} & 2(\Delta x_{n-2} + \Delta x_{n-1}) & \Delta x_{n-1} \\ 0 & 0 & 0 & 0 & \Delta x_{n-1} & 2\Delta x_{n-1} \end{bmatrix} \begin{bmatrix} y''_1 \\ y''_2 \\ y''_3 \\ . \\ y''_{n-1} \\ y''_n \end{bmatrix} =$$

$$\begin{bmatrix} 6\left(\frac{\Delta y_1}{\Delta x_1} - y'_1\right) \\ 6\left(\frac{\Delta y_2}{\Delta x_2} - \frac{\Delta y_1}{\Delta x_1}\right) \\ 6\left(\frac{\Delta y_3}{\Delta x_3} - \frac{\Delta y_2}{\Delta x_2}\right) \\ \cdots\cdots\cdots \\ 6\left(\frac{\Delta y_{n-1}}{\Delta x_{n-1}} - \frac{\Delta y_{n-2}}{\Delta x_{n-2}}\right) \\ 6\left(y'_n - \frac{\Delta y_{n-1}}{\Delta x_{n-1}}\right) \end{bmatrix}$$

The analysis above illustrated the calculation of the cubic spline based upon the values of the spline second derivatives at the nodes based on the equations (I). However, equations (II) show that the cubic spline coefficients can also be calculated from the spline first derivatives at the nodes. In a manner completely analogous to the above discussion, consider equations (II) and the pair of equations (2.3). From these, for k = 1, ··· ,n-1

$$(2.7) \qquad y''_k = 2B_k = \frac{2}{\Delta x_k}\left(3\frac{\Delta y_k}{\Delta x_k} - 2y'_k - y'_{k+1}\right)$$

and

$$(2.8) \qquad y''_{k+1} = 6A_k\Delta x_k + 2B_k$$

$$= \frac{2}{\Delta x_k}\left(-3\frac{\Delta y_k}{\Delta x_k} + y'_k + 2y'_{k+1}\right)$$

and using (2.7) and (2.8) with the requirement for continuity in the second derivative, the following result is obtained:

$$(2.9) \qquad 3\left(\frac{\Delta y_{k-1}}{\Delta x^2_{k-1}} + \frac{\Delta y_k}{\Delta x^2_k}\right) = \frac{y'_{k-1}}{\Delta x^2_{k-1}} + 2\left(\frac{1}{\Delta x_{k-1}} + \frac{1}{\Delta x_k}\right)y'_k + \frac{y'_{k+1}}{\Delta x_k}$$

Similar to (2.6), equation (2.9) is a system of n-2 equations in the n unknowns $y'_1, \cdots ,y'_n$. Here, boundary values for y'_1 and y'_n can be chosen to yield a system of n-2 equations in n-2 unknowns. Again, a situation exists where equations (2.7) and (2.8) can be used to express y'_1 and y'_n in terms of the boundary values y''_1 and y''_n respectively, which results in two additional equations which can be added to the system (2.9) to produce a system of n equations in the n unknowns $y'_1, \cdots, y'_n$.

$$\text{(2.9a)} \qquad \frac{2}{\Delta x_1} y'_1 + \frac{1}{\Delta x_1} y'_2 = \frac{3}{\Delta x_1} \frac{\Delta y_1}{\Delta x_1} - \frac{y''_1}{2}$$

$$\text{(2.9b)} \qquad \frac{1}{\Delta x_{n-1}} y'_{n-1} + \frac{2}{\Delta x_{n-1}} y'_n = \frac{y''_n}{2} + \frac{3}{\Delta x_{n-1}} \frac{\Delta y_{n-1}}{\Delta x_{n-1}}$$

As was the case with the system of equations (2.6), the system of equations (2.9) are tridiagonal and diagonally dominant, guaranteeing a unique solution. Thus, a choice of forms exists with which to determine a one dimensional cubic spline interpolation, each of which allows a completely flexible choice of boundary conditions. The tridiagonal systems of equations lend themselves to a very simplified form of Gaussian elimination. Of course, an n-2 system of equations will generally be quicker to solve, so that the choice between (2.6) and (2.9) will often be determined by the choice of the 'free' or 'clamped' boundary condition. With reference to the bicubic spline, to which this discussion is leading, the cross derivative $\partial^2 S/\partial x \partial y$, must also be continuous at each sampling node. The cross derivative can also be written $\partial/\partial y(\partial S/\partial x)$. To determine the cross derivative at each sampling node, an algorithm which focuses on calculating spline first derivatives is useful, since it can be applied to the 'x' or 'y' partial derivatives at each node to determine the cross derivative at each node. Thus, the form of equations (2.9) will be preferred when determining a bicubic spline interpolant for a 2 dimensional sample set.

2.1.2.3 Bi-cubic spline interpolation

This section describes the extension of the one dimensional cubic spline to two dimensions. Here the one dimensional algorithm described above is used to advantage to determine the coefficients of a bi-cubic spline interpolation for a two dimensional set of data samples.

To describe the two dimensional problem [39], let u_{ij} ($i = 1, \cdots, n$ and $j = 1, \cdots, m$) be the set of known samples of an otherwise unknown and assumed smooth surface $u(x,y)$, evaluated at the nodes of the rectangular grid defined by

$$R: \left\{ \begin{matrix} x_1 < \dots < x_n \\ y_1 < \dots < y_m \end{matrix} \right\}$$

In this region, the rectangles R_{ij} can be defined as

$$R_{ij} = \{(x,y) : x_i \le x \le x_{i+1}, y_j \le y \le y_{j+1}\}$$

and in each of these rectangles the bi-cubic polynomials

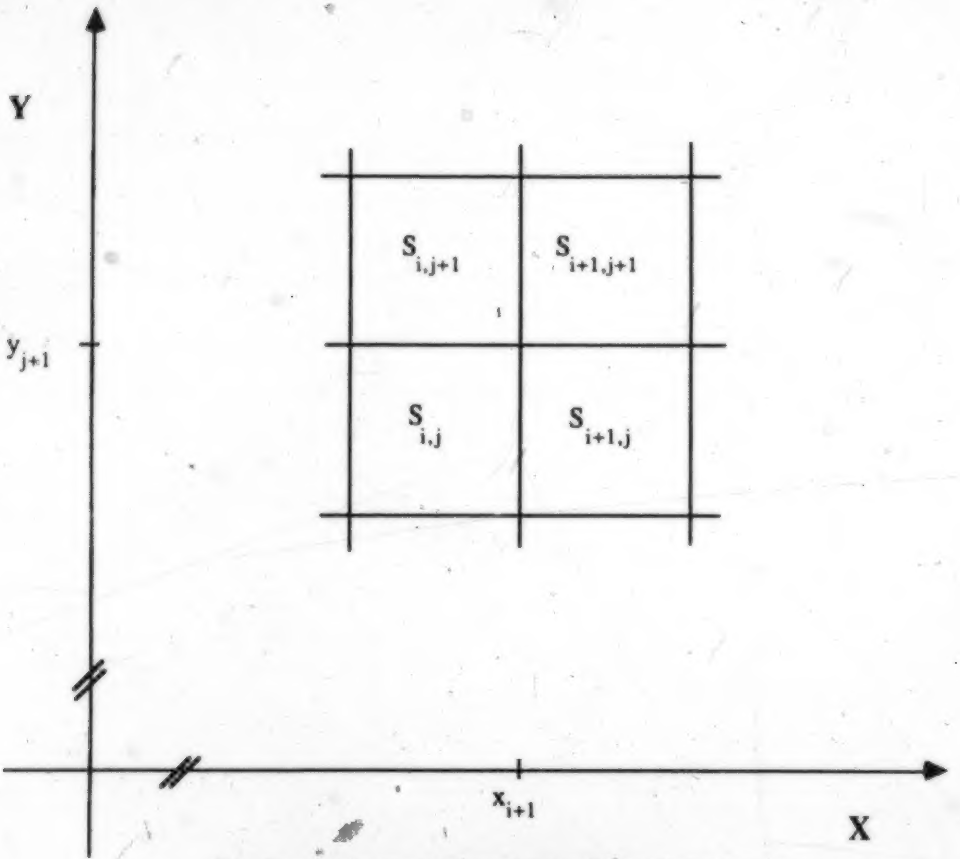
$$S_{ij} = \sum_{k,l=1}^{4} a_{ijkl} (x-x_i)^{k-1}(y-y_j)^{l-1}$$

$$\text{where } i = 1, \dots, n-1 \text{ and } j = 1, \dots, m-1. \tag{2.10}$$

Figure 2.2 and the following equations describe the conditions for two dimensional continuity of the bi-cubic spline polynomials which largely follow from the one dimensional conditions described earlier (in two dimensions, a further requirement is that the cross derivative $\partial^2 S/\partial x \partial y$ be continuous across the nodes of the grid R). At any point (x_{i+1}, y_{j+1}) in the grid R, as described in figure 2.2, the following conditions of continuity apply:

$$\begin{aligned}
S_{i,j}(x_{i+1},y_{j+1}) &= S_{i+1,j}(x_{i+1},y_{j+1}) = S_{i,j+1}(x_{i+1},y_{j+1}) = S_{i+1,j+1}(x_{i+1},y_{j+1}) \\
S^{x}_{i,j}(x_{i+1},y_{j+1}) &= S^{x}_{i+1,j}(x_{i+1},y_{j+1}) = S^{x}_{i,j+1}(x_{i+1},y_{j+1}) = S^{x}_{i+1,j+1}(x_{i+1},y_{j+1}) \\
S^{y}_{i,j}(x_{i+1},y_{j+1}) &= S^{y}_{i+1,j}(x_{i+1},y_{j+1}) = S^{y}_{i,j+1}(x_{i+1},y_{j+1}) = S^{y}_{i+1,j+1}(x_{i+1},y_{j+1}) \\
S^{xx}_{i,j}(x_{i+1},y_{j+1}) &= S^{xx}_{i+1,j}(x_{i+1},y_{j+1}) = S^{xx}_{i,j+1}(x_{i+1},y_{j+1}) = S^{xx}_{i+1,j+1}(x_{i+1},y_{j+1}) \\
S^{yy}_{i,j}(x_{i+1},y_{j+1}) &= S^{yy}_{i+1,j}(x_{i+1},y_{j+1}) = S^{yy}_{i,j+1}(x_{i+1},y_{j+1}) = S^{yy}_{i+1,j+1}(x_{i+1},y_{j+1}) \\
S^{xy}_{i,j}(x_{i+1},y_{j+1}) &= S^{xy}_{i+1,j}(x_{i+1},y_{j+1}) = S^{xy}_{i,j+1}(x_{i+1},y_{j+1}) = S^{xy}_{i+1,j+1}(x_{i+1},y_{j+1})
\end{aligned} \tag{2.11}$$

$$\text{where } S^{x} = \frac{\partial S}{\partial x}, \; S^{xx} = \frac{\partial^2 S}{\partial x^2} \text{ and } S^{xy} = \frac{\partial^2 S}{\partial x \partial y}.$$



Continuity conditions for the Bi-cubic spline

Figure 2.2

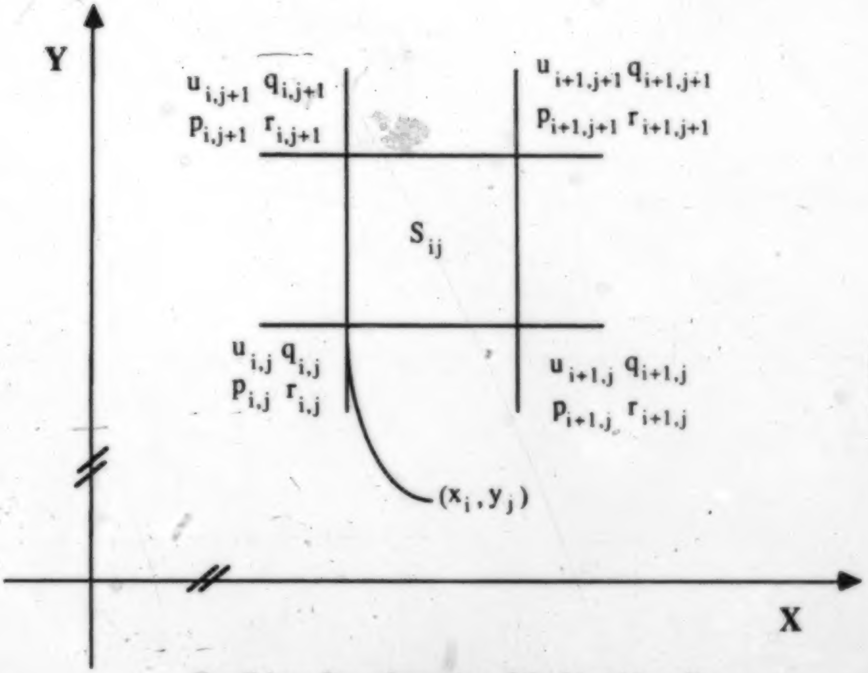
Examination of (2.10) and the continuity conditions (2.11) shows that any 'x' or 'y' cross section of a bi-cubic spline is itself a one dimensional cubic spline and that a cross section across a grid line in R is the one dimensional cubic spline interpolating those points. The following abbreviations will be used:

$$p_{ij} = \frac{\partial S_{ij}(x_i,y_j)}{\partial x}$$

$$(2.12) \qquad q_{ij} = \frac{\partial S_{ij}(x_i,y_j)}{\partial y}$$

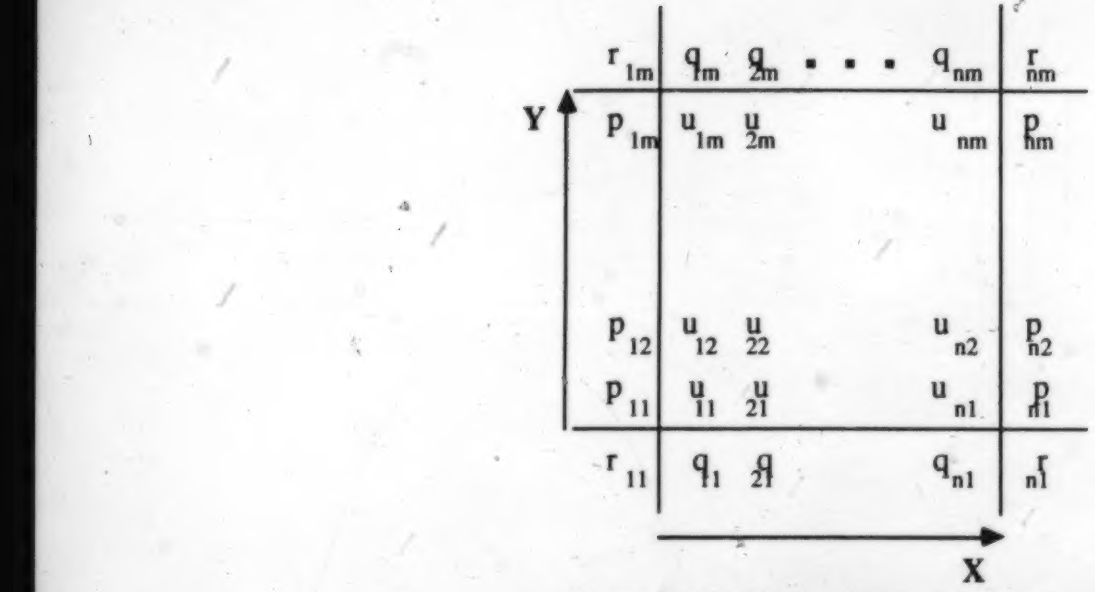
$$r_{ij} = \frac{\partial S_{ij}(x_i,y_j)}{\partial x \partial y}$$

Examination of figure 2.3 shows that if the values of u,p,q and r are known at each corner of a given bi-cubic spline polynomial S_{ij} a condition exists where 16 known values can be used to solve for the 16 unknown coefficients of S_{ij}.


Y
$u_{i,j+1}$ $q_{i,j+1}$
$p_{i,j+1}$ $r_{i,j+1}$
$u_{i+1,j+1}$ $q_{i+1,j+1}$
$p_{i+1,j+1}$ $r_{i+1,j+1}$
S_{ij}
$u_{i,j}$ $q_{i,j}$
$p_{i,j}$ $r_{i,j}$
$u_{i+1,j}$ $q_{i+1,j}$
$p_{i+1,j}$ $r_{i+1,j}$
(x_i, y_j)
X

Conditions for uniqueness of the bi-cubic spline
Figure 2.3

The discussion above focused upon the general case of interior sections of a sampled image, but the method of determining the values p,q and r has not been discussed. In general, two dimensional images encountered in practice are bounded, and the number of boundary conditions sufficient to interpolate the sampled image unknown. Figure 2.4 illustrates the boundary conditions needed for a general 'n x m' bounded image [39].



Necessary boundary conditions to uniquely solve the bi-cubic spline

Figure 2.4

To determine the values p_{ij}, q_{ij} and r_{ij}, one dimensional splines are first fitted to the rows and columns of u_{ij} to determine the values of p_{ij} and q_{ij} respectively. Here, either the 'free' or 'clamped' boundary condition can be used as desired. Next, the grid of r_{ij} values ($\partial^2 S/\partial x \partial y$) need to be evaluated. To begin, a one dimensional cubic spline is determined for the first and last rows of q_{ij} (an alternative procedure would be to fit a one dimensional cubic spline to the columns of p_{ij}). Doing this assumes either a 'clamped' or 'free' boundary value for r_{ij} at each corner. These are the last boundary value assumptions needed to determine a unique bi-cubic spline. Now, that the r_{ij} values for the first and last row have been evaluated they can be used as 'clamping' values for cubic splines determined for the columns of p_{ij} (or in the alternative procedure, the 'clamping' values for the rows of q_{ij} will have been determined); ie:- the 'clamped' boundary value algorithm must now be used , to fit the one dimensional splines to the columns of p_{ij}.

From the procedure described above, grids for p_{ij}, q_{ij} and r_{ij} can be determined, and with the original knowledge of the u_{ij} values, 16 simultaneous equations are all that remain to be evaluated for each Sij to determine their associated coefficients and thus fully express the bi-cubic spline interpolating function desired.

2.2 Digital Filtering techniques

Digital filter design has undergone significant development in recent years, providing the signal processing engineer with a useful and flexible tool. Digital filters fall into one of two broad categories: (1) Infinite Impulse Response (IIR) filters and (2) Finite Impulse Response (FIR) filters. "Although a strong association exists between IIR filters and recursive realizations, and between the association of FIR filters and nonrecursive realizations, these associations are not absolute. It has been shown [25] that both IIR and FIR filters can be implemented by either recursive or nonrecursive techniques"[41]. However, IIR filters generally employ recursive realizations, and FIR filters typically have nonrecursive realizations. FIR filters are also commonly realized using the Discrete Fourier Transform (DFT). Both of the IIR or FIR filtering techniques have particular advantages and disadvantages as will be discussed.

2.2.1 IIR filter design

A very common approach to IIR filter design is based on the considerable body of techniques established for 'continuous time linear time invariant' (CTLTI) systems [32,41]. The first step in this process is to find a low-pass prototype filter which meets specified design requirements. When this has been done, transformation techniques are established [41] to convert this low-pass equivalent to the desired 'band pass', 'band reject' or 'high pass' CTLTI prototype. Once the CTLTI prototype transfer function has been obtained, it can be mapped from the continuous or Laplace transform domain to the discrete or z-transform domain with the aid of a bi-linear transformation [32,41].
Once the desired 'discrete time linear time invariant' (DTLTI) transfer function has been established, the filter is typically realized in one of a number of recursive forms which implement the difference equation associated with the z-transform transfer function. The major advantage of the IIR or 'recursive' type filter is that fewer delay elements are required to realize a given magnitude response characteristic. However, there are major disadvantages to this type of filter which must be considered:

(1) IIR filters have non-linear phase response characteristics which are aggravated as filter order is increased to achieve faster magnitude response roll-off characteristics.

(2) The design process for IIR filters is more complex than for their FIR counterparts.

(3) The potential for instability is increased as the requirement for improved magnitude response is sought. This is caused by an increasing number of poles close to the unit circle. This can be aggravated, as the computational limits of the hardware are pressed to resolve the space between a z-plane pole and the unit circle.

(4) Also related to the computational limits of the implementing hardware, is it's ability to differentiate between z-plane poles.

2.2.2 FIR filter design

A number of design techniques have been developed for the design of FIR filters, which have no basis in CTLTI system design. This is likely due to the ease of implementing discrete time convolution. A common and straight forward design technique which will be discussed here, is called the 'Fourier Series Method'.

With the 'Fourier Series Method', Fourier series are used to determine samples, at the system sampling rate, of the impulse response of an 'ideal' magnitude response model. However, because the impulse response of the 'ideal' model is infinite, an appropriate means must be found to truncate it to a finite number of terms so that it can be dealt with practically. This truncation process can be viewed as looking at the infinite impulse response through a window of finite width. If the window is rectangular (all coefficients of the impulse response within the window receive equal weighting) then taking the Fourier transform of this truncated impulse response shows considerable 'ringing' or 'side lobes' in the resulting magnitude response. Of course, the larger the window through which the impulse response is viewed (or truncated), the more closely the resulting magnitude response will approximate the 'ideal'.

A number of techniques have been developed to produce window functions which smooth the truncation of the impulse response, with the purpose of reducing the side lobe response in the resulting magnitude response[41]. While these methods reduce the side lobes of the frequency response characteristics by reducing the number of high frequency components introduced into the impulse response by the windowing process, a price is paid in the effective broadening of the main lobe of the resulting frequency response.

The FIR filter design process described above has a number of useful advantages in that they

(1) can have a linear phase response characteristic,
(2) are always stable, and
(3) have a much simpler design process than their IIR counterparts.

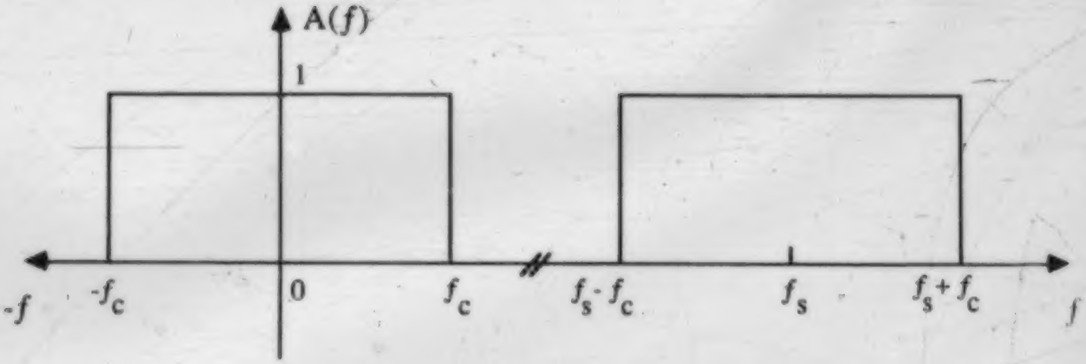
The major disadvantage of the FIR filter is that a much larger number of delay elements are needed to realize a magnitude response with a specified roll-off characteristic as compared with their IIR counterparts. However, the advantages of the FIR filter out weigh the disadvantages, particularly at the system development stage. Thus, the FIR was the filter of choice. In a topographic mapping system for the EEG, digital filters would be used to bring out the desired rhythmicity. At this stage, a system approaching real-time performance may well benefit from the lesser phase delay of an IIR filter, but the non-linear phase response characteristics must be remembered. Thus, the IIR filter does bear future consideration. For now, FIR filter design details are considered.

2.2.2.1 FIR filter details

To understand the Fourier Series Method of FIR filter design, an important starting point is to note that the amplitude response of an ideal DTLTI filter is a periodic function in frequency. Just as periodic time domain signals have discrete frequency domain descriptions, so periodic frequency domain signals have discrete time domain descriptions. Consider an ideal low pass filter transfer function $H(f) = A(f)e^{j\text{Ø}(f)}$ illustrated in figure 2.5, where $\text{Ø}(f) = 0$ for all f. Since $A(f)$ is an even function of frequency, it can be described by a Fourier cosine series in the time domain [41].

$$A(f) = \frac{a_0}{2} + \sum_{m=1}^{\infty} a_m \cos(2\pi m T f) \qquad (2.13)$$

$$\text{where } T = \frac{1}{f_s}$$



Frequency response for an ideal low-pass digital filter

Figure 2.5

The coefficients a_0 and a_m can then be determined as

$$a_0 = \frac{2}{f_s} \int_{-f_c}^{f_c} A(f)\, df = \frac{2}{f_s} \int_{-f_c}^{f_c} df = \frac{2}{f_s} \left[2f_c \right]$$

(2.14)
$$a_m = \frac{2}{f_s} \int_{-f_c}^{f_c} A(f) \cos (2\pi\, m\, T f)\, df = \frac{2}{f_s} \int_{-f_c}^{f_c} \cos (2\pi\, m\, T f)\, df$$

$$= \frac{2}{\pi\, m} \left[\sin (2\pi\, m\, T f_c) \right]$$

$A(f)$ can also be expressed in exponential form as

$$A(f) = \sum_{m=-\infty}^{\infty} c_m\, e^{j2\pi\, m\, T f}$$

(2.15) where

$$c_m = c_{-m} = \frac{a_m}{2} = \frac{1}{f_s} \int_{-f_c}^{f_c} A(f) \cos (2\pi\, m\, T f)\, df$$

The solution to a_m described in equation (2.14) can be put into the form K sin x/x, which is the well known sinc function. To perfectly describe the ideal amplitude response $A(f)$, we require an infinite number of coefficients a_m. Practically, of course $A(f)$ can only be approximated with a finite number of terms, so the approximating expression for $A(f)$ becomes

$$A_1(f) = \sum_{m=-M}^{M} c_m e^{j2\pi m T f} \qquad (2.16)$$

Substituting the z-transform variable $z=e^{j2\pi Tf}$ into equation (2.16), a new series is obtained, and the close relationship between the Fourier and Z transforms can be seen. This series, represented in equation (2.17) below has the form of a discrete transfer function. However, the transfer function is in the form of a two sided Z transorm; that is, it is noncausal because it contains positive powers of z, which in turn represent negative time terms in the impulse response.

$$H_1(z) = \sum_{m=-M}^{M} c_m z^{m} \qquad (2.17)$$

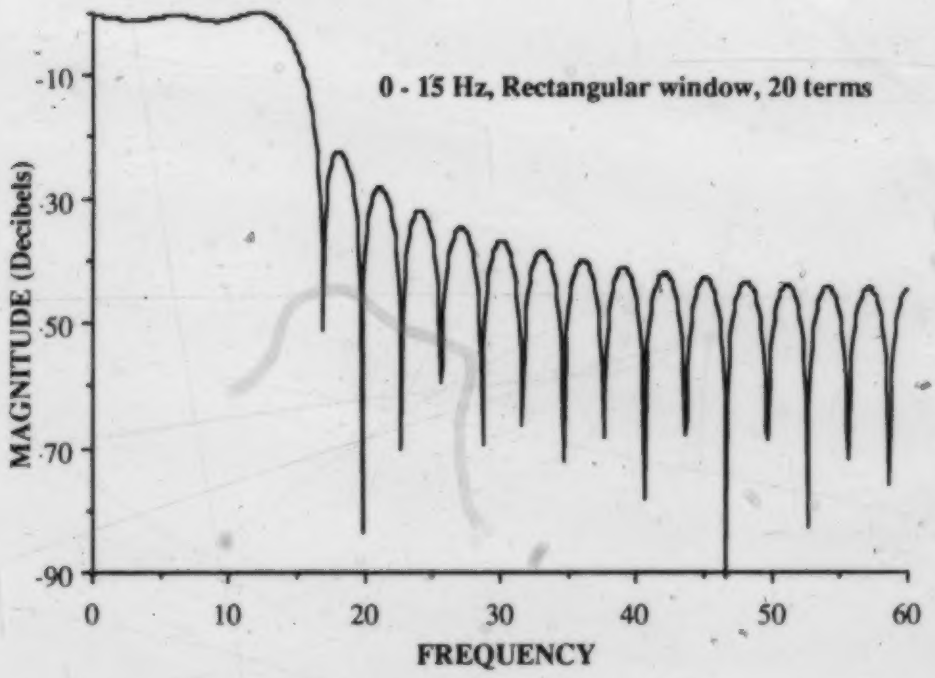
The noncausality of $H_1(z)$ can be dealt with by introducing a sufficient time-delay; that is $H_1(z)$ can be multiplied by z^{-M} to produce a new transfer function, which has the familiar form of a one sided Z transform.

$$H(z) = z^{-M} H_1(z) = \sum_{i=0}^{2M} d_i z^{-i} \qquad (2.18)$$

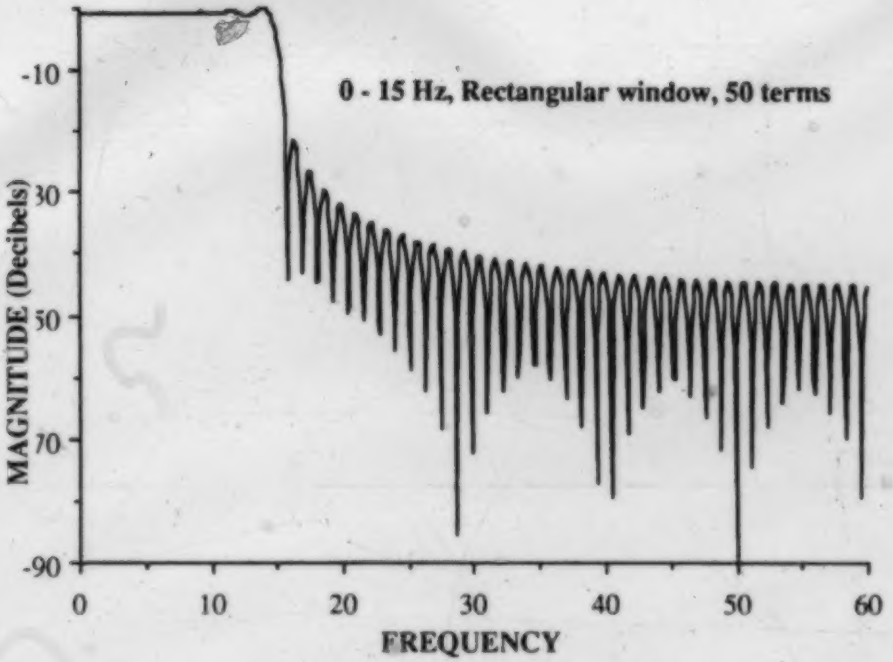
$$\text{where } d_i = c_{M-i}$$

The above expression for H(z) represents an FIR transfer function of order 2M. The set of coefficients d_i are simply time shifted versions of the set of coefficients c_m as can be seen from 2.18 above. Since $Z^{-1}[z^{-m}] = \delta(t - m)$, it is apparant that the Fourier transform coefficients c_M are simply those of the FIR filter impulse response. In the beginning a zero phase response was specified, but the introduction of M time delays (MT) to create a causal transfer function H(z) results in a linear phase response. This time delay does not affect the amplitude response, it does introduce a phase shift $\beta(f) = -2\pi MTf$ which is a linear function of frequency representing a constant time delay of MT seconds.

To illustrate the performance of FIR filters, consider the digital approximation of an ideal low pass filter, which has a specified cutoff frequency of 15 Hz., and where the sampling rate is 120 Hz. Figure 2.6 shows the performance of the low pass filter, when the ideal impulse response is truncated after 20 terms (that is, the filter incorporates 40 delay elements) while figure 2.7 shows the performance of the filter when the ideal impulse response is truncated after 50 terms. These figures illustrate, that for a given filter specification, a sharper roll off characteristic can be obtained in the approximation by simply including more terms from the ideal impulse response characteristic. However, using more terms from the ideal impulse response introduces more delay terms into the digital filter, because the value of M is increased in equation 2.18 above. For a real time hardware implementation of such a filter, this implies an increased phase delay.



Low pass filter 0 - 15 Hz.
Rectangular windowed to 20 terms
Figure 2.6



Low pass filter 0 - 15 Hz.
Rectangular windowed to 50 terms
Figure 2.7

Another noteworthy effect illustrated by figures 2.6 and 2.7 is that due to the abrupt truncation of the impulse response. The truncation process can be viewed as multiplying an infinite impulse response by some windowing function of finite extent, in the time domain. In this case, the windowing function is a rectangular window, which multiplies impulse response values inside the window by 1 and values outside the window by 0 (that is, the impulse response function has been abruptly truncated). This rectangular window has an amplitude frequency response of infinite duration (the sinc function) and since multiplication in the time domain is equivalent to convolution in the frequency domain, this window response has been convolved with the ideal frequency response model to produce the actual amplitude response illustrated by figures 2.6 and 2.7. The amplitude response resulting from the above process, is unsatisfactory in a sense due to the presence of relatively large side lobes. These side lobes, as was alluded to above, are due to the sharp discontinuity at the edge of the window response.

A number of windowing functions have been found [41] which smooth the truncation near the window boundary. The frequency response of a smooth window function has significantly reduced side lobes compared with that of a rectangular window (the sinc function). However, the main lobe of a smooth window function is broader than that of the sinc function. Thus, while the use of windowing functions can significantly suppress side lobe response, main lobe response can be significantly broadened. Some notable examples of window functions are:

Hanning window:

$$
\begin{aligned}
w(t) &= \frac{1}{2}\left(1 + \cos \frac{2\pi t}{\tau} \right) \qquad \text{for } |t| \leq \frac{\tau}{2} \\
&= 0 \text{ elsewhere}
\end{aligned}
\tag{2.19}
$$

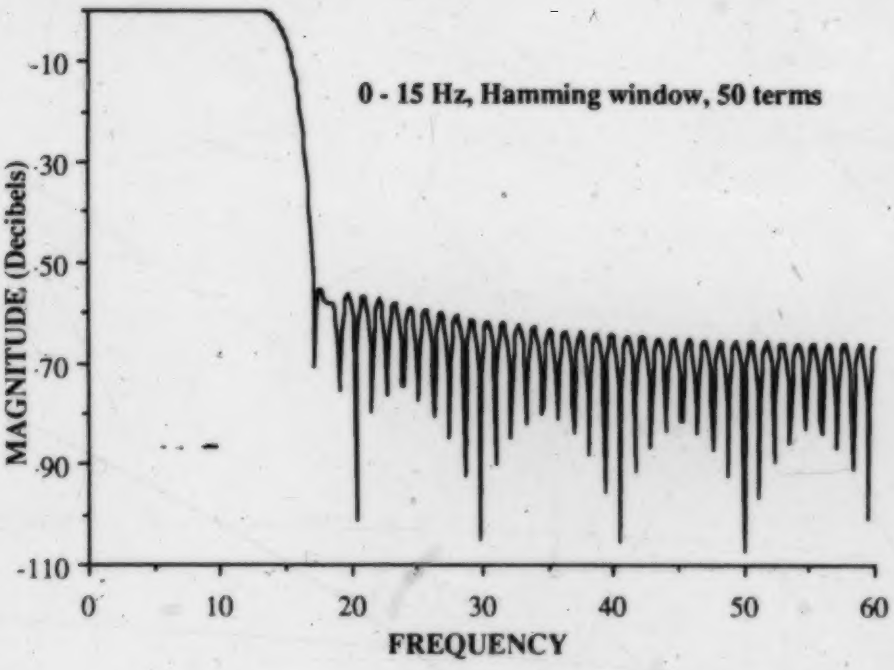
Hamming window:

$$
\begin{aligned}
w(t) &= 0.54 + 0.46 \cos \frac{2\pi t}{\tau} \qquad \text{for } |t| \leq \frac{\tau}{2} \\
&= 0 \text{ elsewhere}
\end{aligned}
\tag{2.20}
$$

Kaiser window:

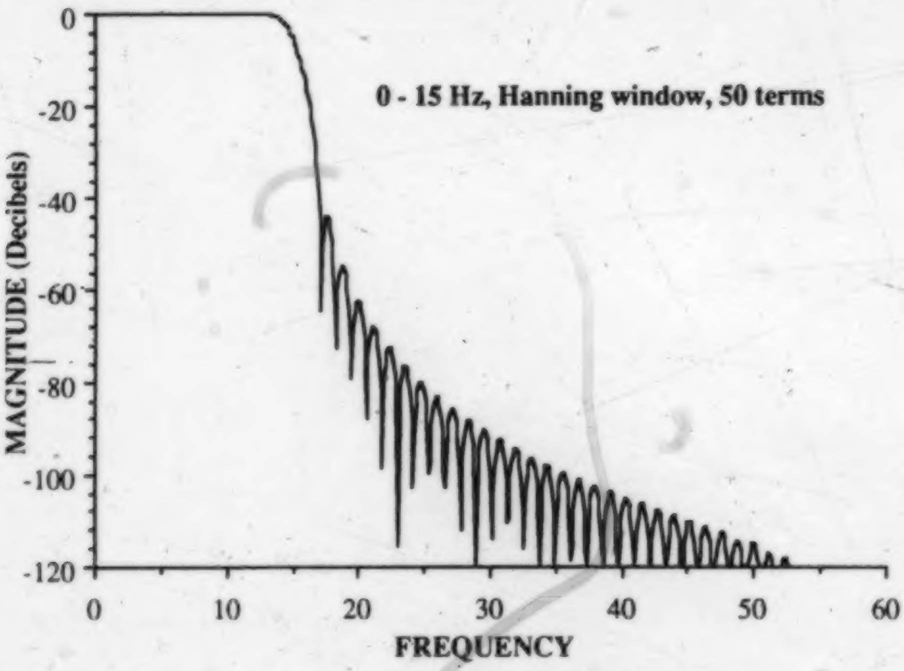
$$
w(t) = \frac{I_0\left[\theta \sqrt{1 - (2t/\tau)^2} \right]}{I_0(\theta)} \qquad \text{for } |t| \leq \frac{\tau}{2}
\tag{2.21}
$$

In the above expressions, τ represents the window width 2MT. For the Kaiser window, I_0 denotes the modified Bessel function of the first kind of order zero, and θ is an adjustable parameter called the shape factor. The shape factor allows for a tradeoff between main lobe width and side lobe height. Larger shape factors correspond to wider main lobes and smaller side lobes. The Kaiser window has the best potential for side lobe suppression [41], of the above three window functions. However, ready access to computer library routines which evaluate the modified Bessel function may not be available to some designers. The Hamming window provides about 5 dB better side lobe suppression in the first side lobe. However, side lobe suppression as frequency increases further begins to level off at about -52 dB. On, the other hand, side lobe suppression of the Hanning window continues to improve as frequency increases.



Low pass filter 0 - 15 Hz.
Hamming windowed to 50 terms
Figure 2.8

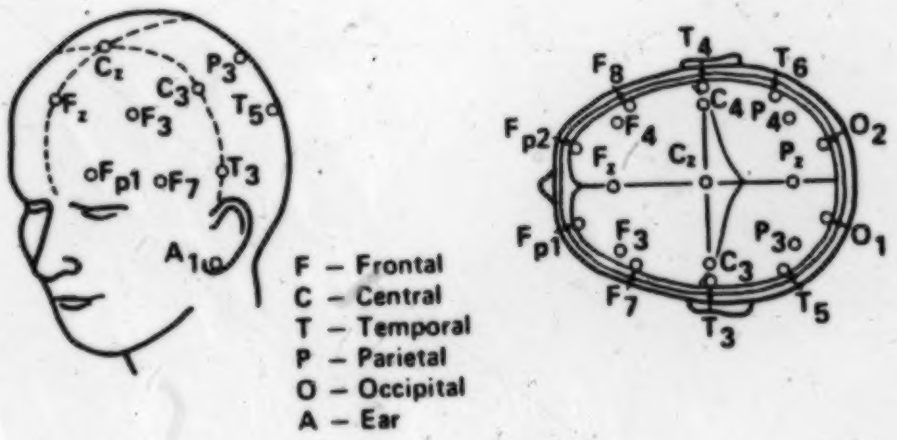
Figures 2.8 and 2.9 describe the improvements to the 15 Hz. low pass filter described above, where the Hanning and Hamming window functions respectively truncate the ideal impulse response after 50 terms, ie: $\tau = 2MT = 2(50)T = 100\ T$.



Low pass filter 0 - 15 Hz.
Hanning windowed to 50 terms
Figure 2.9

2.3 The Electroencephalogram

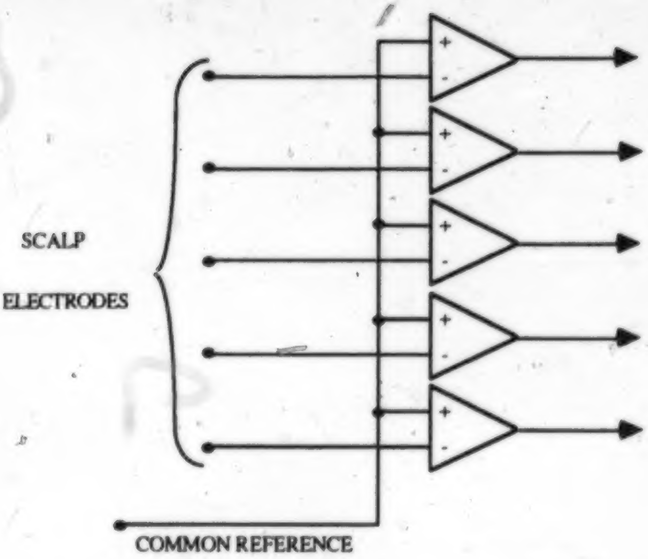
Clinical recordings of the electroencephalogram (EEG) are obtained by attaching electrodes to particular locations on the scalp of the subject being studied: traditionally, the electric potentials measured by these electrodes are presented to the clinician in strip chart form. The manner of presentation of the EEG has been considered by many researchers to be a major limiting factor in the usefulness of the EEG. Topographic mapping, a central aspect of this report, is a relatively recent development which addresses this major issue. However, a more fundamental issue affecting the clinician's understanding of the EEG is the selection of an electrical reference. The electrical reference issue, and the manner in which this project approached its resolution, will now be considered.


Cz
P3
C3
Fz
F3
T5
Fp1
F7
T3
A1
F – Frontal
C – Central
T – Temporal
P – Parietal
O – Occipital
A – Ear
T4
F8
T6
C4
P4
Fp2
F4
O2
Fz
Cz
Pz
Fp1
F3
P3
O1
C3
F7
T5
T3

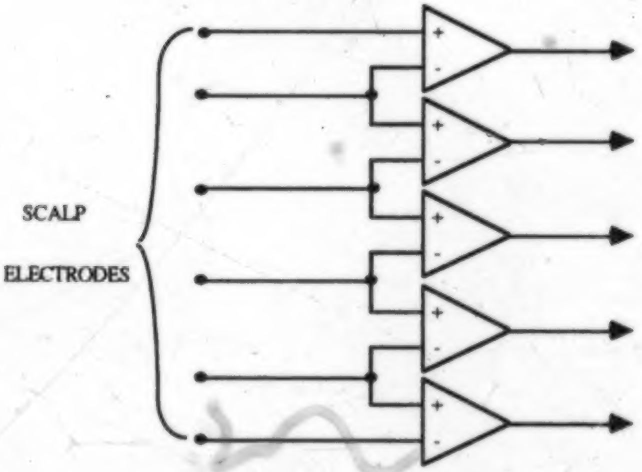
The International 10 - 20 System
of electrode placement
Figure 2.10

2.3.1 Recording configurations

There are two basic arrangements or montages commonly used to record the EEG; the bipolar montage and the unipolar montage. With the unipolar montage, electrodes are arranged over the scalp and their potentials relative to a chosen electrical reference are recorded. A standard for electrode locations, is specified by the International 10-20 System. This system attempts to locate 19 electrodes uniformly over the scalp, and locates 1 or 2 additional reference electrodes at the ears, as illustrated in figure 2.10. With the bipolar montage, the potential differences between pairs of the electrodes are recorded. Figure 2.11 illustrates the differences between these 2 recording configurations.


SCALP
ELECTRODES
+
-
COMMON REFERENCE

(a) Unipolar Configuration


SCALP
ELECTRODES
+
-

(b) Bipolar Configuration

Recording configurations

Figure 2.11

2.3.1.1 Bipolar recordings

The bipolar recording method is inherently a reference 'free' recording method. Here, the voltage difference resulting from scalp currents is recorded. Based on simplified electrical models of the scalp- skull-brain system, the bipolar montage has been thought to have the ability to detect scalp currents caused by cortical generators local to the electrode locations [29]. However, when these simplified models begin to take into account openings in the highly resistive skull (skull resistivity is approximately 80 times greater than for scalp and brain [35]), it can be seen that scalp currents due to generators remote to the recording sites, but passing through skull openings, can be comparable to currents due to generators close to recording sites but passing directly from the generator through the skull to the scalp. Bipolar recordings may also be subject to polarity reversal, due to small lateral shifts of the electrode positions[29], depending upon the physical nature of the cortical generator involved.

It is difficult to imagine how the bipolar configuration could be employed in a topographic mapping system since these signals are vector quantities proportional to tangential current flow in the scalp, assuming ohmic conduction. In other words, the bipolar configuration represents a sampled 2 dimensional image of potential gradients. It is difficult to represent such a gradient image in 2 dimensions on a color monitor because of the continuously variable directional information present in such an image. Thus, while the bipolar montage avoids the electrical reference issue, it is quite impractical for topographic mapping.

2.3.1.2 Unipolar recordings

The unipolar montage measures scalp potential variations relative to a common reference potential, and so is a reference dependant recording configuration. These scalp potentials are also easily interpolated spatially into topographic maps of scalp potential. Ideally, the common reference would be a point far from any sources of potential. Choosing a reference off the subject is impossible because of the high impedance in such a measurement loop relative to the input impedance of the measurement amplifiers. In addition, such a measurement system would only introduce noise from environmental sources, corrupting the relatively low scalp potentials to be recorded. Thus, it becomes necessary to choose a reference on the subject under study. However, if a reference on the body is chosen, it is impossible to find an electrically quiet reference, due to potentials generated within the body. Thus measured scalp potential variations would always be relative to the potential variations of the reference. If the body is not placed close to external sources, the potentials induced on the body by these environmental sources tends to be constant over the body surface [29], which means that EEG electrodes will not be sensitive to these potentials since they are distributed evenly over the body surface.

In the modern era of the digital computer, the unipolar montage is a very flexible measurement configuration. Digitizing the measured EEG makes practical the use of a multitude of waveform analysis techniques, one of which is coherent averaging, which are difficult to implement or even imagine with earlier analog techniques. The bipolar montage can be implemented as desired by remembering that the simple arithmetic relationship $(\Phi_1 - \Phi_R) - (\Phi_2 - \Phi_R) = \Phi_1 - \Phi_2$ which removes the effect of the reference potential to provide a potential difference measurement as in the bipolar configuration.

It can be seen from the above discussion, that while the unipolar configuration may be preferred in studying the EEG, placement of the reference electrode has a significant effect on the potentials recorded. Careful consideration must be given to the placement of the reference electrode, its effect on EEG measurements, and how these effects might be minimized or eliminated.

2.3.2 The reference electrode

It can be shown that placing a reference electrode at the neck is essentially equivalent to placing a reference electrode anywhere else on the body below the neck [29] except for the introduction of EKG artifact (which can easily be prevented from inducing large currents in the scalp by placing a conductive collar around the neck). Thus the search for a suitable placement for the reference electrode becomes restricted to the head itself, making it difficult to distinguish between scalp and reference electrodes.

The skull, which is much more resistive than scalp or brain tissue, has a marked effect on how currents flow in the scalp. There is a tendency for currents generated by sources within the brain to be concentrated at holes in the skull as they attempt to follow paths of lower resistance. At this point then, it can be seen that the goal of obtaining an electrically inactive reference is impossible. The question which might be asked at this point is: "is it possible to minimize or remove the effect of the reference electrode?".

Generators of electrical activity in the brain, particularly in the cerebral cortex, can often be modeled effectively as dipoles, or dipole sheets [35]. These generators in turn give rise to currents flowing in the scalp. In this case, it may make more sense to study scalp currents rather than scalp potentials, as indicators or brain electrical activity. Studying scalp currents also provides an advantage, since they are independant of the electrical reference.

2.3.2.1 The Laplacian

The EEG is a measure of the spatial distribution of electric potential over the scalp, which at any instant in time can be expressed as $(\Phi - \Phi_R)$ where Φ_R is the reference electrode potential at the moment of interest. From the potential distribution, it is a straight forward matter to determine the scalp current density as

$$J = -\sigma\nabla(\Phi - \Phi_R) = -\sigma\nabla\Phi$$

where σ is the scalp conductivity and ∇ is the well known gradient operator. However, J is a vector quantity, and as such is difficult to represent in the form of a topographic map, since both magnitude and direction of current flow need to be described.

At any instant in time, regions of high cortical activity can be viewed as dipole sheets [29]. These high activity areas will be 'sourcing' current upward through the skull and scalp while the areas of lower activity will be 'sinking' current downward through the scalp and skull. These radial currents can serve to better identify local areas of cortical activity. Tangential currents, on the other hand, such as those expressed by J above do not identify 'source' and 'sink' areas.

The scalp can be considered as a two-dimensional conductive sheet overlying a two-dimensional poorly conducting shell. For the purposes of this work, this poorly conducting shell, the skull, will be assumed to be of uniform thickness and resistivity, simplifying this analysis. Since scalp current must be conserved, the 'source' and 'sink' areas described above can be determined by evaluating the two-dimensional divergence of the horizontal vector field J [29]:

$$\nabla \bullet J = - \nabla \bullet (\sigma\nabla\Phi) = -\sigma\nabla^2\Phi$$

or

$$-\frac{1}{\sigma}\nabla \bullet J = \nabla^2\Phi = \frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2}$$

The right hand side of this last equation is the Laplacian of the scalp electric potential, is independent of the reference potential and is a scalar quantity. If the Laplacian can be applied to the two dimensional interpolation of the scalp potential surface, a topographic map can be produced representing 'source' and 'sink' current activity. In addition to being reference 'free', such maps could also serve to better locate cortical generators.

Chapter 3
SYSTEM DEVELOPMENT AND PERFORMANCE

Having considered the theoretical basis for the topographic mapping system of this report, details of the system implementation will now be reviewed. The material which follows is divided into two sections covering hardware and software aspects respectively.

The topographic mapping system, has two major components; one for data acquisition and one for system processing. Data obtained with the acquisition system, was digitized and stored on magnetic media for later processing on the mapping system hardware. The creation of topographic maps results in considerable data expansion and so includes large volume magnetic media in the form of a 60M byte 1/4 inch streaming tape drive for storing topographic maps.

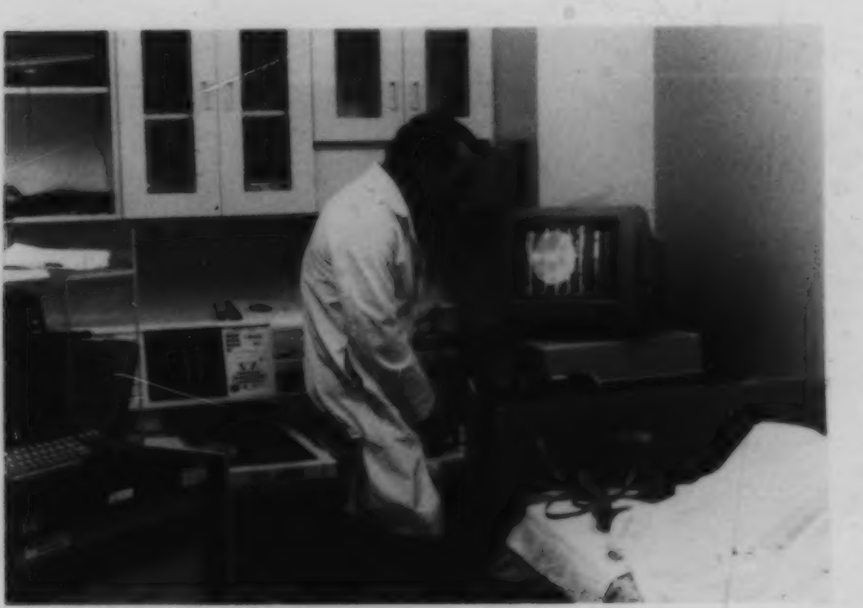
The major thrust of this project was to produce software for creating 'reference free' topographic maps. The programming languages used in system development were FORTRAN77 and Motorola 68000 ASSEMBLER. Within the Department of Applied Sciences in Medicine, subroutine libraries were developed for interfacing with the various graphics, storage and signal processing hardware attached to the microprocessor system bus. Appendix A contains descriptions of these subroutines.

3.1 System hardware

As mentioned earlier, system hardware in the project had two major components; data acquisition hardware and mapping system hardware. Data acquisition hardware consisted of a stand alone 32 channel digitizing system developed by Dr. Zoly Koles in the Department of Applied Sciences in Medicine. Input to the digitizing system was supplied via the amplifier array of a Neurosciences Brain Imager topographic mapping system. The topographic mapping system for the project reported here, was developed on a Motorola VME 10 microprocessor system and its associated hardware.

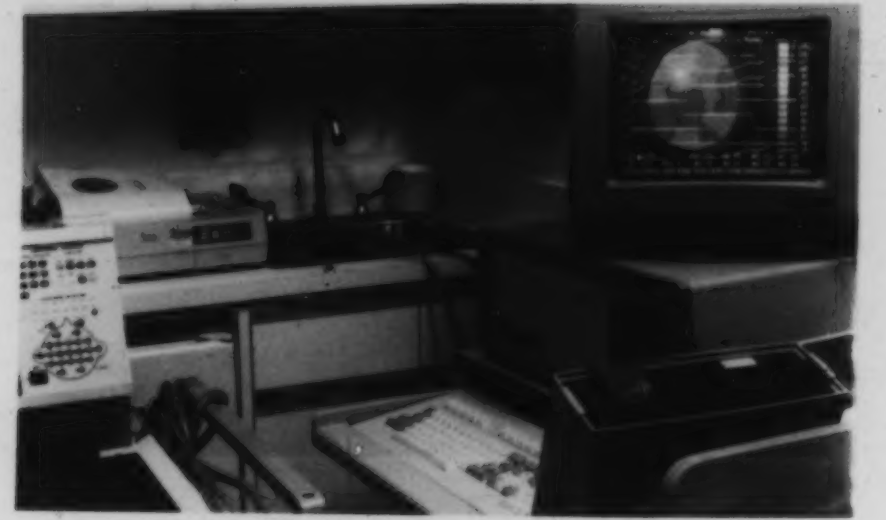
3.1.1 Data Acquisition

In chapter 1, it was indicated that a number of topographic mapping systems are available commercially. In the Department of Neurology at the University of Alberta Hospital, where all of the EEG recordings were made for this project, the Neurosciences Brain Imager was available for use. The main purpose for using this machine in our project was for acquiring data via its 31 electrode analog recording amplifier array. EEG data was recorded using a single ear reference unipolar configuration illustrated in figure 2.10; 31 electrodes were arranged in the scheme of figure 1.2 and 1 reference electrode was placed on the ear. The 31 channel data thus obtained formed the input to the digitizing system where it was stored on 8 inch floppy disks. Figures 3.1 through 3.3 show a series of photographs depicting the recording environment.

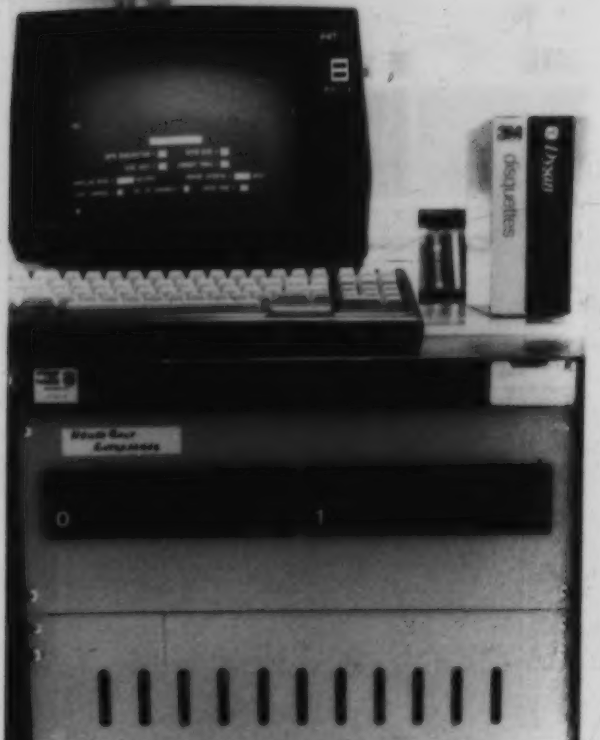


A typical recording session

Figure 3.1



The Neurosciences machine
Figure 3.2



disquettes
Dysan

The digitizing system
Figure 3.3

The Neurosciences Brain Imager mentioned above is typical of systems which are currently commercially available. From the point of view of this report, the basic disadvantages of this machine are: 1) the maps produced are reference dependant potential maps 2) the 28 electrodes of the system effect a less dense spatial sampling of the scalp potential surface and 3) the interpolation method used for map generation is of a lower order than that used in this project. The Neurosciences machine does however incorporate a number of useful features, but since the use of this machine was not crucial to this project other than in the manner indicated, further discussion of it will not be included.

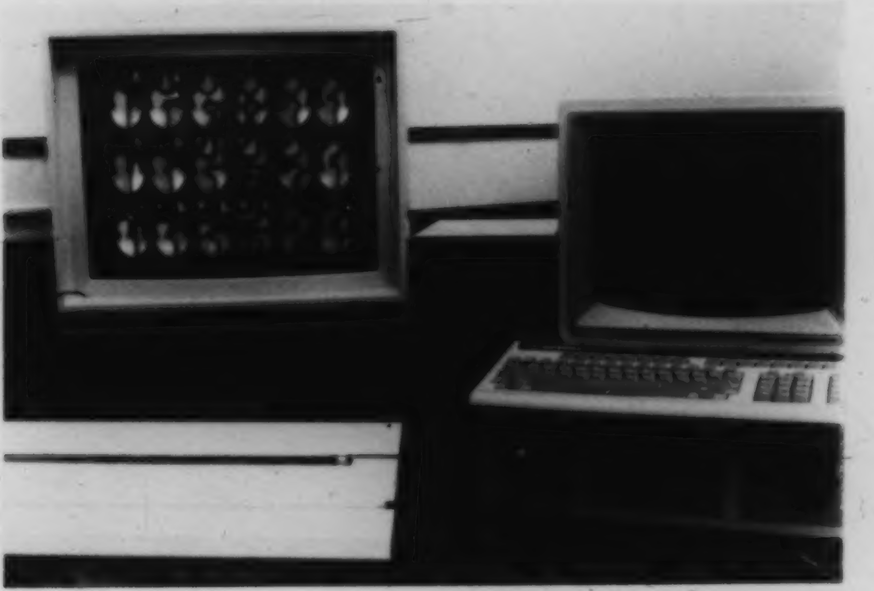
The digitizing system discussed above was built around the Texas Instruments 990/101MA microcomputer and the ANALOG DEVICES RTI - 1240/1241 digitizing peripheral which is memory mapped to the TI 990 micro computer. Each analog channel is time division multiplexed to the digitizing circuitry of the RTI 1240 with a cycle time between conversion of each channel of about 40µ sec. For the mapping system of this report, this cycle time implies a time skew between the digitizing of channels 1 and 32 of 1.28 msec. which is quite significant relative to the system sampling period of 8.3 msec. (120 samples/sec.). However, as was mentioned in chapter 2, the highest frequency signal for which accurate and precise interpolations could be produced was about 15 Hz. Relative to the 66.7 msec. period of a 15 Hz. signal, the 1.2 msec. skew mentioned above, though undesirable, was not considered prohibitive.

3.1.2 Mapping hardware

The heart of the topographic mapping system hardware for this project is the Motorola VME 10 microcomputer system, based on an MCM 68010 16/32 bit microprocessor and the VME bus. Attached to the VME bus are a number of second source board level products which provide 2M bytes of RAM expansion, fast floating point arithmetic, digital signal processing, color graphics for topographic mapping and 1/4 inch format streaming tape storage / retrieval

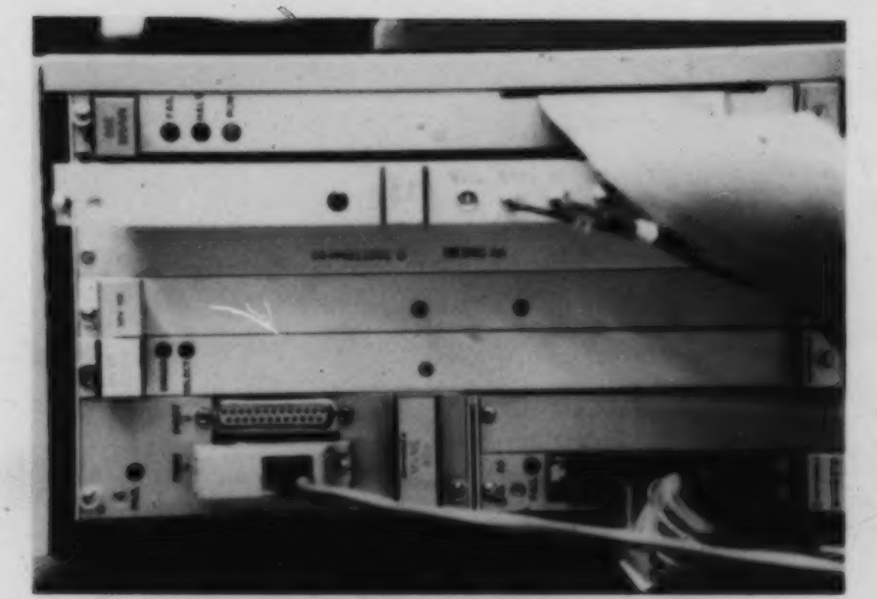
3.1.2.1 VME 10 microcomputer

As mentioned above, the basic processing unit of this computer system is the MCM 68010 microprocessor, which is upwardly compatible with newer versions in the 68000 family of microprocessors. The 68010 microprocessor and its successors are part of the most recent generation of microprocessors, able to bring to bear powerful addressing, data handling, speed and arithmetic capabilities in support of high level language development for a modern topographic mapping system. The VME bus specification is an asynchronous specification which evolved from bus development work by Motorola for the 68000 family of devices [28]. Today, the VME bus is a well defined specification capable of data transfer rates between 20M and 40M bytes/sec. [28,42] and has received wide international manufacturing acceptance. At present, the are some 100 manufacturers which provide board level products supporting the VME bus, to provide support for any conceivable application. The VME 10 microcomputer system in the Department of Applied Sciences in Medicine is configured with a number VMEbus modules supporting the topographic mapping project of this report and is illustrated in the photographs of figures 3.4 and 3.5. The functional features of these boards are described in the following paragraphs:



Front view of the VME 10 development system

Figure 3.4



Rear view of the VME 10 development system
showing VMEbus modules

Figures 3.5

Memory expansion

A characteristic of image processing is the frequent manipulation of large memory arrays. Because of the large memory requirement, the system memory was expanded with a single board product from Motorola, which contained 2M bytes of dynamic RAM directly addressable by the processor via the VME bus and equal in performance to memory on the system board. This memory expansion brought the total system RAM to 2.7M bytes.

Floating point arithmetic

Operation of this topographic system required many floating point operations due to the large dynamic range of numerical values encountered during its operation. To be able to perform these operations quickly, a fast floating point processor (FFP) board from SKY Computers Inc., was configured into the system via the VME bus and a library of subroutines which interface with the board. These subroutines are known to the system linker, and supported by the FORTRAN77 compiler resident on the VME 10 computer system. The SKY floating point processor board makes available 104 single and double precision floating point operations.

Because of the overhead involved in calling the SKY floating point subroutines, benchmark testing was performed to fairly compare the speed of the SKY floating point operations with the standard FORTRAN floating point operations which used the host processor. Multiplication on the SKY FFP was about 3 times faster while the calculation of certain transcendental functions was about 20 times faster than using the 68010 instruction set for these operations. Since most of the time complexity in the mapping program involved the 4 standard arithmetic operations, overall system performance of about 50% was realized with the introduction of the FFP.

The arithmetic hardware for the SKY FFP consists of four 2901 bit-slice processors and a high speed 16 by 16 multiplier and operates as a slave device to the system bus. Communication to and from the host processor is performed through seven 16 bit programmed I/O registers.

Image storage and retrieval

Topographic maps generated by this mapping system are color images containing 15,000 picture elements (pixels). Images can be stored as either sets of 2 byte integers representing numeric values for the scalp potential (or scalp radial current density) surface, or sets of nibble integers representing color scaling values for the scalp potential (radial current density) surface. In the former case, the storage requirement is 30,000 bytes per topographic map. Since the mapping system can generate a map for every temporal sampling interval (120 per second), the storage requirement for every map in a 20 second interval is 72M bytes. Configured in the VME 10 system are a 15M byte hard disk and a 5 1/4 inch double density format floppy disk drive, which do not provide sufficient storage capability given the requirements mentioned above.

An MVME 350 streaming tape controller board and 60 Mbyte 1/4 inch format streaming tape drive were purchased from Motorola and Archive Corporation respectively, for mass storage purposes. The MVME 350 is a VME bus compatible product providing the industry standard QIC-02 REV D intelligent tape drive interface between the host VME 10 system and the Archive Scorpion model 5945 1/4 inch streaming tape drive which has a data transfer rate of 90 Kbytes per second.

Color graphics display

Capability for color graphics display of topographic maps is provided by a Gigatec Ltd. color graphics monitor and a DY-4 Corporation DVME 778 raster graphics card. The DVME 778 is a VME bus compatible product implemented using the NEC 7220 graphic display controller IC device. This board will allow the selection of 16 colors out of a palette of 4096 colors for display at any one time, via look up tables. Video RAM is configured to a display format of 640 x 480 pixels and memory mapped to the host VME 10 computer system on a write only basis.

Digital signal processing

This capability is provided via a Burr-Brown SPV 100 signal processing board. The SPV 100 utilizes a Texas instruments TMS 320 digital signal processor and is equipped with firmware to perform a 256 point FFT. The firmware for the FFT utilizes an in place integer arithmetic type algorithm which allows 4 input/output options:

1) Real input - Magnitude output
2) Real input - Complex output
3) Complex input - Complex output
4) Complex input - Magnitude output

The SPV 100 performs a 256 point FFT in 7.2m sec.

Program memory on the SPV 100 for the TMS 320, can be selected as either a 4k 16 bit words of static RAM or 4k 16bit words of PROM. Thus, program RAM or PROM can be loaded with TMS 320 executable code to perform the desired digital signal processing application. As mentioned above, the SPV 100 on this system was equipped with PROM to perform a 256 point FFT. A TMS 320 cross assembler was not available for the VME 10 system at the time of the project, so that custom design of digital signal processing applications was not performed to utilize the full program capabilities available on the SPV 100.

Data memory on the SPV 100 is organized as two 4k 16 bit words of static RAM. At any time, the host CPU on the VME bus has access to one 4k x 16 buffer while the TMS 320 has access to the other. These two sets of data memory are handled as swinging buffers allowing dual port access to one memory while the other is being processed.

3.2 System software

The major thrust of the topographic mapping system reported here is software. This system software has two major parts: the first deals with producing and storing topographic maps of the EEG and is called WRTAPE while the second deals with retrieval and display of those topographic maps and is called RDTAPE. The basic form of WRTAPE and RDTAPE along with major design considerations will be discussed. Both WRTAPE and RDTAPE were written in FORTRAN77.

The objectives of WRTAPE were to 1) temporally filter EEG data as desired 2) spatially interpolate between the scalp electrode values at each instant in time to produce a continuous topographic map illustrating either the reference dependant scalp potential or the reference independent scalp radial current density and 3) display and store these images for each sampling instant as they were produced. Early efforts to produce these topographic maps required a cycle time to display successive images of 2.5 min. With the introduction of the SKY fast floating point processor and assembly language code designed to optimize the evaluation of these maps, the cycle time was reduced to 21 sec. However, it still would take about 12 hours of processing to produce topographic maps for about 16 seconds of EEG data if maps were created for every sampling interval. Thus it was necessary to create RDTAPE for the effective viewing of EEG data. RDTAPE allows the user to review topographic maps at any point in a 16 second long data file in a number of different formats to be discussed in detail.

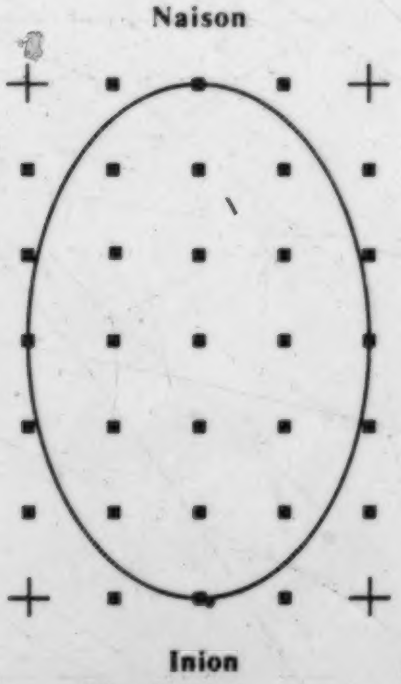
As mentioned earlier in this report, a Library of Assembly and FORTRAN utility subroutines were developed by a number of individuals during the course of this project, for which descriptions are in appendix A. Appendix B contains listings and descriptions of FORTRAN subroutines obtained from Spath [39] which were used to perform cubic and bi-cubic spline interpolation. These subroutines in appendix B reflect some changes made by the author to implement bi-cubic spline interpolation based upon the 'free' spline boundary condition as well as the 'clamped' spline boundary condition discussed in chapter 2. Appendix C contains photographic plates of typical color graphics displays generated on the color graphics monitor by WRTAPE and RDTAPE.

3.2.1 WRTAPE

3.2.1.1 Creating the maps

The electrode array of figure 1.2, for the purposes of making topographic maps, was projected on to the planar array of figure 3.6 and to this array the bi-cubic spline interpolation method was applied to produce topographic maps. An ellipse would be drawn within the electrode array to create a map shape generally indicative of head shape from a coronal perspective. However, before the bi-cubic spline could be applied, some decision had to be made to select a value for the 4 missing corners of the rectangular array needed to apply the bi-cubic spline. It was decided to allow the user the flexibility of selecting either a zero value at the corner, or a bi-linear average of the values from the 3 nearest neighbors.

Next, a decision had to be made as to how large the final topographic map was to be. If the map was too small, it would be inconvenient to view; the larger the map, the more pixel values would be required, increasing the time complexity of the program. Early experimentation in the project, interpolating 5 by 5 arrays indicated that a 128 by 128 pixel topographic map was as small as could be accepted from a visual perspective. However, the electrode array to be interpolated was now a 5 by 7 array so that some adjustment was in order. During project development it was found that an effective means for displaying the maps was sequentially in a left to right manner from top to bottom, just as people read a page of text and as illustrated in figure 3.7. The dimensions of the color graphics display monitor attached to the system was 640 pixels horizontally by 480 pixels vertically. If the map size was taken to be 100 by 150 pixels, 3 rows of 6 topographic maps could be displayed on a single screen, leaving enough room at the bottom of the screen for comments and a color bar. Thus each topographic map to be created would require the evaluation of 15000 pixel values.

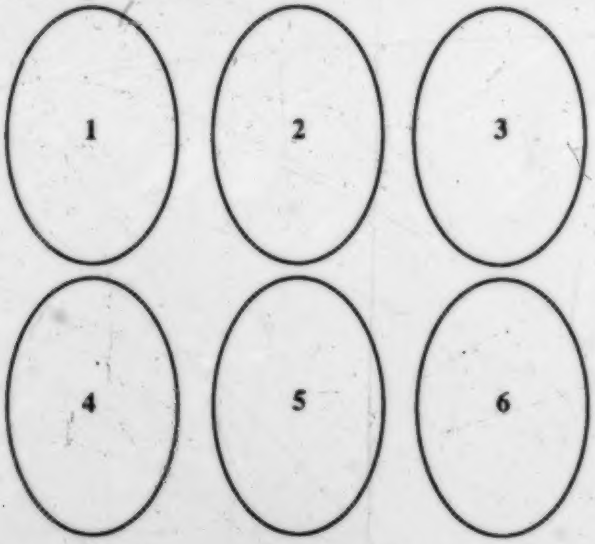


Mapping array

Figure 3.6

3.2.1.2 Storing Topographic maps

The size of the topographic maps would in turn affect how many maps could be stored on a single 60M byte tape cassette. An additional factor in determining an answer to this question was a decision to write a single block header file at the beginning of the tape, followed by a raw data file from which the suceeding maps were made. Each point in the topographic map would be scaled to one of 16 color values so that maps could be compressed into a 'nibble' format to conserve storage space. This meant that each map could be stored in 7500 bytes, corresponding to 15 tape blocks @ 512 bytes per block. One sample interval for 31 channels of raw data would require 64 bytes of storage. One 512 byte block would be used for storing header information at the beginning of the tape. Based on these factors, it was found that 1940 topographic maps could be stored on one tape along with 1 block of header information and 1 raw data file corresponding to the raw data from which all 1940 images were made.



Display configuration

Figure 3.7

3.2.1.3 Evaluating topographic maps

In chapter 1, it was pointed out that the major goal of this project, was to create topographic maps which would be independent of the electrical reference used to record the passive EEG. In the last part of chapter 2, it was shown how the application of the Laplacian operator to the scalp potential surface would yield a description of radial currents which were independent of the electrical reference. As has been discussed, the scalp potential surface was to be approximated with bi-cubic spline polynomials. That is, to each rectangular region between the electrode locations illustrated in figure 3.6, a bi-cubic polynomial was to be determined according to the conditions described in chapter 2, which would be of the form

$$
\begin{aligned}
\Phi(x,y) = \sum_{i=1}^{N-1}\sum_{j=1}^{M-1} & a_{11} + a_{21}(x - x_i) + a_{31}(x - x_i)^2 + a_{41}(x - x_i)^3 \\
& + a_{12}(y - y_j) + a_{22}(x - x_i)(y - y_j) + a_{32}(x - x_i)^2(y - y_j) + a_{42}(x - x_i)^3(y - y_j) \\
& + a_{13}(y - y_j)^2 + a_{23}(x - x_i)(y - y_j)^2 + a_{33}(x - x_i)^2(y - y_j)^2 + a_{43}(x - x_i)^3(y - y_j)^2 \\
& + a_{14}(y - y_j)^3 + a_{24}(x - x_i)(y - y_j)^3 + a_{34}(x - x_i)^2(y - y_j)^3 + a_{44}(x - x_i)^3(y - y_j)^3
\end{aligned}
$$

where N and M are the number of horizontal and vertical samples

The Laplacian would be applied analytically to these polynomials to yield interpolating polynomials of the form

$$
I(x,y) = -\nabla^2\Phi_{ij}(x,y) = -\left[\frac{\partial^2\Phi_{ij}}{\partial x^2} + \frac{\partial^2\Phi_{ij}}{\partial y^2}\right]
$$

$$
\begin{aligned}
= \sum_{i=1}^{N-1}\sum_{j=1}^{M-1} & 2a_{31} + 6a_{41}(x - x_i) \\
& + 2a_{32}(y - y_j) + 6a_{42}(x - x_i)(y - y_j) \\
& + 2a_{33}(y - y_j)^2 + 6a_{43}(x - x_i)(y - y_j)^2 \\
& + 2a_{34}(y - y_j)^3 + 6a_{44}(x - x_i)(y - y_j)^3 \\
& + 2a_{13} + 2a_{23}(x - x_i) + 2a_{33}(x - x_i)^2 + 2a_{43}(x - x_i)^3 \\
& + 6a_{14}(y - y_j) + 6a_{24}(x - x_i)(y - y_j) + 6a_{34}(x - x_i)^2(y - y_j) + 6a_{44}(x - x_i)^3(y - y_j)
\end{aligned}
$$

Thus, topographic maps could be produced to illustrate either the reference dependant scalp electric potential or the reference independent scalp current densities.

3.2.1.4 Scaling the topographic maps

Raw EEG data are digitized by the data acquisition system in 12 bit 2's complement format, and stored as 2 byte 2's complement integer values. However, for any given raw EEG data file, no prior knowledge of actual values existing in the file is available. Yet, such knowledge is necessary for scaling values in the topographic map to one of the 16 color values available for display. In addition, each map must be scaled within a single range in order to depict variations in the relative intensity of each map. To facilitate choosing an appropriate scaling value, code was included in WRTAPE which would count the number of occurrences of each of the possible 2048 possible magnitudes, and display them as a histogram. From this histogram, a scaling value could be chosen appropriate for most of the file.

3.2.1.5 Filtering the data

Chapter 2 included a discussion of the various ways in which filters can be constructed. It was decided that for the purposes of this project, the zero phase response characteristics and the ease of construction of FIR filters made them ideal for the development process. The user was allowed the flexibility of choosing the desired pass band frequencies. The ideal impulse response function for each filter was to be truncated after 50 terms and windowed with the Hamming window function. The only justification for using the Hamming window rather than the Hanning window was that the former has a bit better supression of the first sidelobe and slightly better roll off performance. The performance of filters for the various pass bands is illustrated in figure 3.8.

The filters were implemented using discrete convolution of the time domain data. This would mean that the first and last 50 samples would not yield a valid output due to end effects and thus topographic maps could only be produced from 50 time delays after the beginning of the file to 50 time delays prior to the end of the data file.

3.2.1.6 Program flow

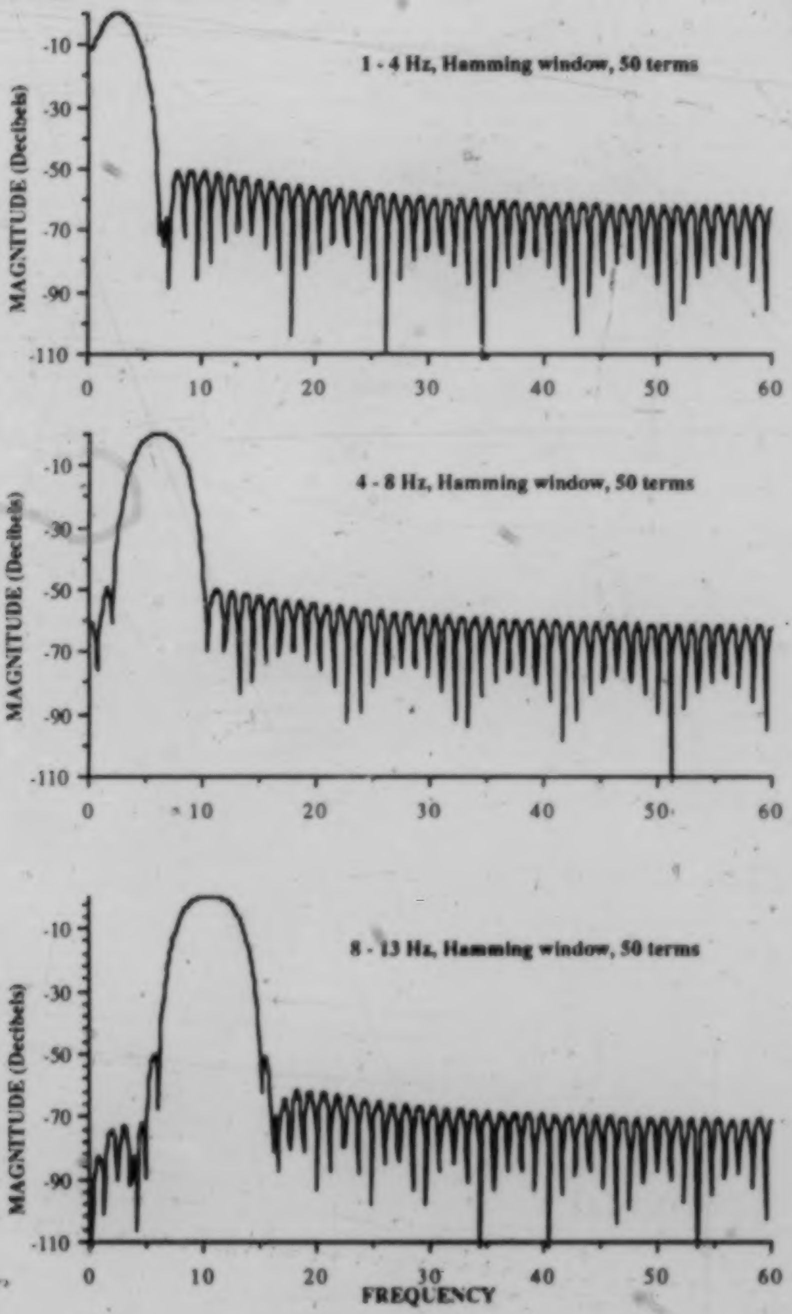
The basic program flow for WRTAPE is illustrated in figure 3.9. The following is a step by step explanation of the program features depicted in this flow chart.

Get raw data

To begin making topographic maps of a subject EEG, raw data must first be made available on the VME 10 computer system's hard disk drive. The user is prompted for the appropriate filename, which is saved and written to a header file along with other information such as program options the user wishes to select or other descriptive information. Having obtained the filename, the system reads 2040 frames of raw data (31 channel samples per frame) into the system memory where it is organized into 31 columns of 2040 samples each.

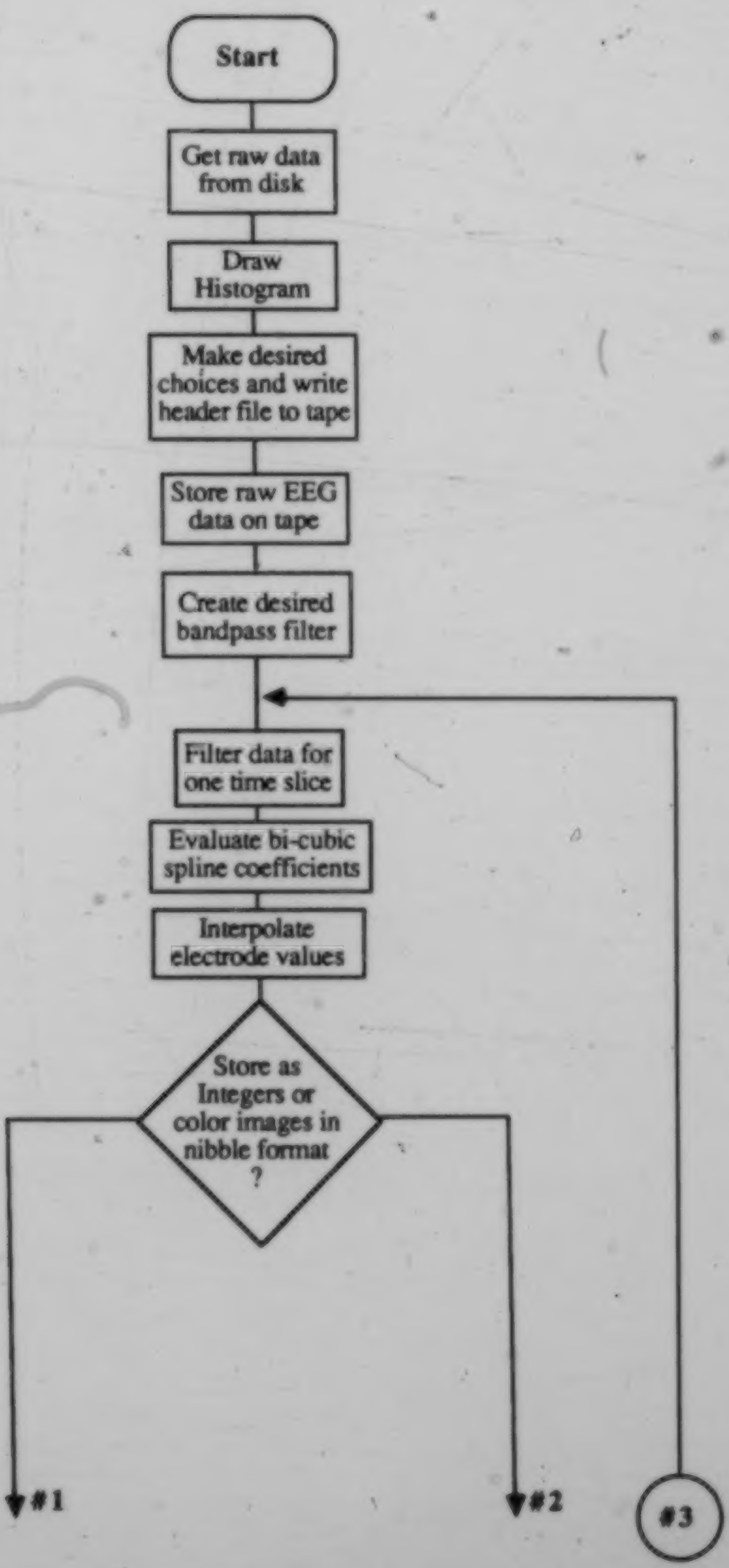
Draw Histogram

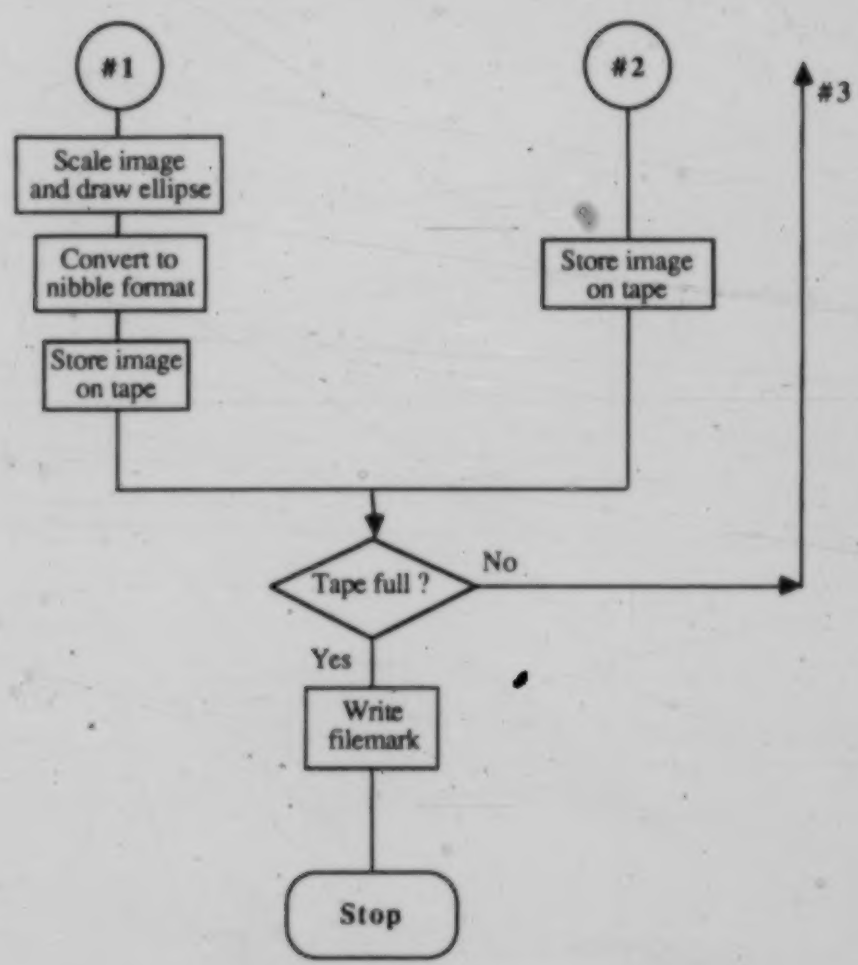
The number of occurrences of each possible sample magnitude is recorded, and the information displayed on the color graphics monitor in the form of a histogram. From this histogram, the user can select which magnitude limits to use for scaling to the 16 available color values. Normally, the user would not want to scale to the maximum



Frequency response plots of Delta band, Theta band and Alpha band filters

Figure 3.8


Start
Get raw data from disk
Draw Histogram
Make desired choices and write header file to tape
Store raw EEG data on tape
Create desired bandpass filter
Filter data for one time slice
Evaluate bi-cubic spline coefficients
Interpolate electrode values
Store as Integers or color images in nibble format ?
#1
#2
#3


#1
Scale image and draw ellipse
Convert to nibble format
Store image on tape
#2
Store image on tape
#3
Tape full ?
No
Yes
Write filemark
Stop

WRTAPE flowchart

Figure 3.9

recorded sample value since most data in the file is below this level and would be scaled at much too low a level, thus reducing important detail.

Header file

The user is allowed to select a number of optional values on which WRTAPE operates. These values are relevant to the interpretation of the topographic maps produced. The user is also prompted for other descriptive details which may be relevant to the interpretation of maps. The header file is 512 bytes of character data corresponding to one physical block on the storage tape. This header file can be customized as desired. The following is a list of the options available to the user:

1) Filter cutoff frequencies
2) Scaling value selected from the histogram
3) Selection of 0 or a bi-linear average for the corner positions in the rectangular electrode array
4) Selection of either potential or current maps
5) Selection of either the 'free' or 'clamped' boundary condition for the bi-cubic spline
6) Topographic maps can be stored on tape as 2 byte integers corresponding to actual interpolated values (in which case fewer than 1940 maps can be produced) for later statistical processing. Alternatively, topographic maps can be scaled to create color images and converted to nibble format.

Other descriptive information which the user may wish to include as relevant to the interpretation of the topographic maps are:

1) Raw data filename
2) Patient name
3) Patient handedness
4) Recording date
5) Patient Test scores
6) General remarks

This information can be customized as desired.

Store raw data

It is considered that the user will likely wish to compare topographic maps with the raw data from which they came. Thus, the raw data from which the maps were made is stored on the magnetic tape after the header file. Before the raw data is stored (and later processed), the spatial average at each time slice or frame is extracted from each channel to remove the D.C. component. This has no effect on the current maps, as they are reference independent. Similarly, the voltage maps will not be affected except to change the reference voltage to an average reference voltage, but the resulting display of the raw data traces by RDTAPE will be improved with the removal of large D.C. fluctuations.

Create bandpass filter

It was discussed in chapter 2 that for the purposes of topographic mapping, temporal examination of the EEG must be restricted to frequencies below 15 Hz. Within this frequency range, 3 classic EEG bands are of interest to the clinician; delta band (1 - 4 Hz.), theta band (4 - 8 Hz.) and alpha band (8 - 13 Hz.). These frequency ranges are somewhat nominal. Typically, a clinician evaluating an EEG is required to quantify activity in these 3 bands visually, while digital signal processing techniques make it possible to accurately and consistently quantify frequency information. Thus, WRTAPE provides the user with the opportunity to construct an appropriate digital bandpass filter, to spectrally decompose the EEG to the rhythm of interest. The performance and implementation of these filters has been discussed in chapter 2.

Filtering

At this stage, the repetitive portion of WRTAPE is entered which will create topographic maps for successive frames. Filtering is the first stage of this operation. The filter kernel is centered at the time of interest, overlaying the raw data to perform discrete convolution, for each channel in succession. The filtered values so obtained for each channel, comprise the array of values to be interpolated spatially.

Evaluating spline coefficients

With the filtered array of electrode values complete, the coefficients (a_{ij}) for the bi-cubic spline polynomials can be determined according to the algorithm discussed in chapter 2. The user will have selected, in the choices section described earlier, either the 'free' or 'clamped' boundary condition.

Interpolating the electrode grid

Having determined the coefficients for the bi-cubic spline polynomials, it is a straight forward matter to determine the interpolated values for each of the 100 x 150 points in the topographic map. This is the most time complex portion of WRTAPE, where the bi-cubic polynomials must be evaluated for each of the 15000 points in a topographic map. The result, is a 100 x 150 array of 2 byte integer values.

Each polynomial evaluation requires 14 floating point additions, 18 floating point multiplications and 1 floating point to integer conversion. After optimizing the evaluation routine in 68010 assembler and using the SKY FFP, evaluation of one topographic map has been reduced to 19 seconds out of a total cycle time of 21 seconds per map.

Storing the maps

As indicated above, the result of the interpolation process is a 100 x 150 array of 2 byte integer values. The user has been given the option of storing this array directly, or converting it to a scaled elliptical color image to be stored in nibble format. The integer array option allows the user the flexibility to develop and perform a variety of statistical operations which may be of interest, before the mapping process is complete. The storage requirement for the integer array is 59 physical tape blocks as opposed to 15 physical tape blocks for the color maps stored in nibble format.

Finally a test is made to determine if the tape is full of images, or whether the mapping process should be repeated. According to the tape format outlined above, 1940 color maps can be stored on a 60M byte tape, representing 16.17 seconds of EEG record.

3.2.2 RDTAPE

The design decisions that were made regarding the construction of topographic maps and their storage have been discussed above. Due to the time requirement associated with interpolation, the prospect of real time processing of these maps remains in the future. In the meantime, a facility was needed to separately retrieve and review maps in order to make the most effective use of the user's time. (A tape of maps made with WRTAPE, at the present stage of development requires 11.3 hours of processing time.) Thus RDTAPE provides the user with information about how the recordings and maps were made, the ability to visually review raw EEG and the ability to review topographic maps either as a page of 18 sequential maps or a movie style replay presentation of enlarged maps.

3.2.2.1 Program flow

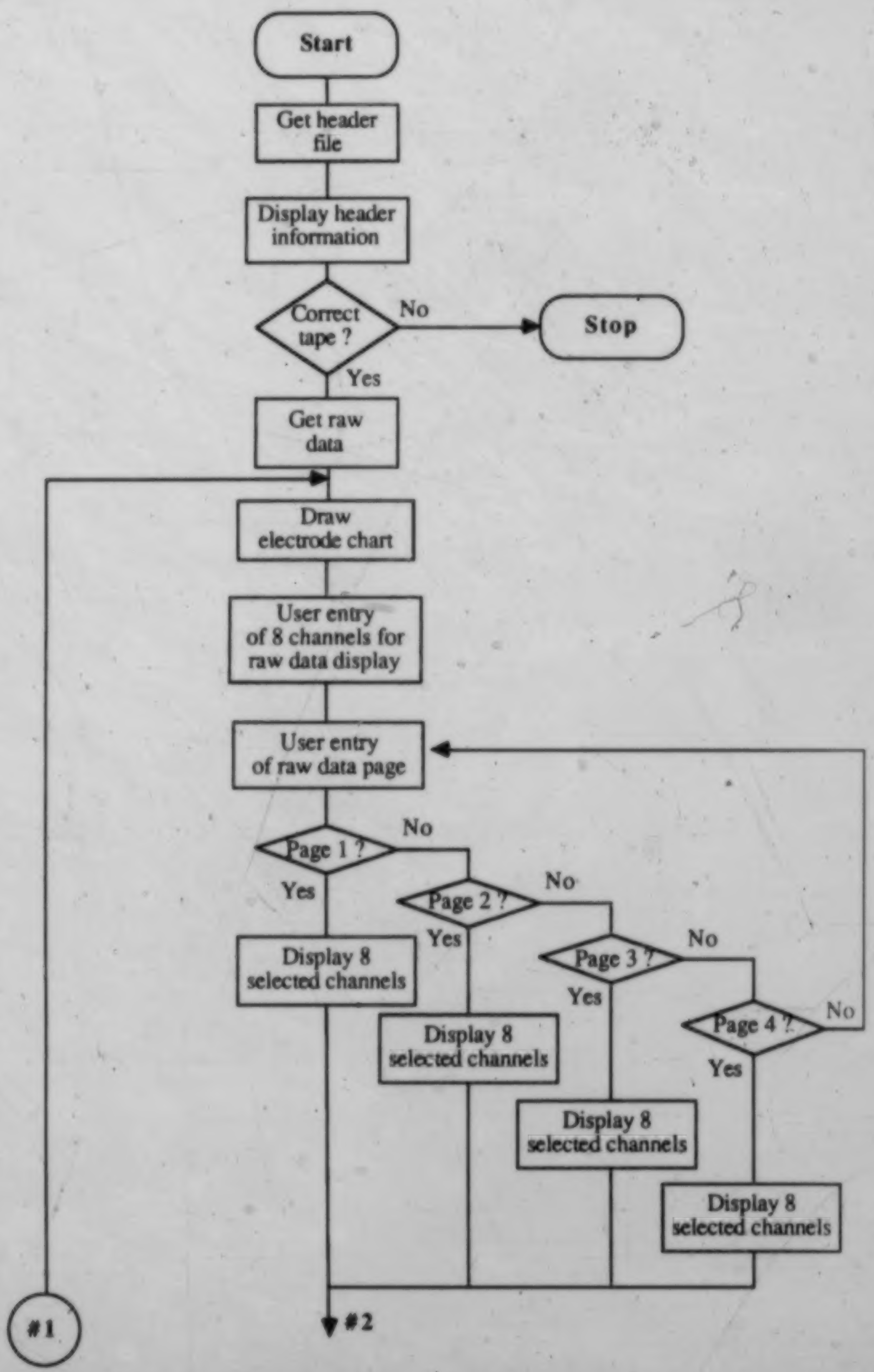
The basic program flow for RDTAPE is illustrated in figure 3.10. The following is a step by step explanation of the program features depicted in this flow chart.

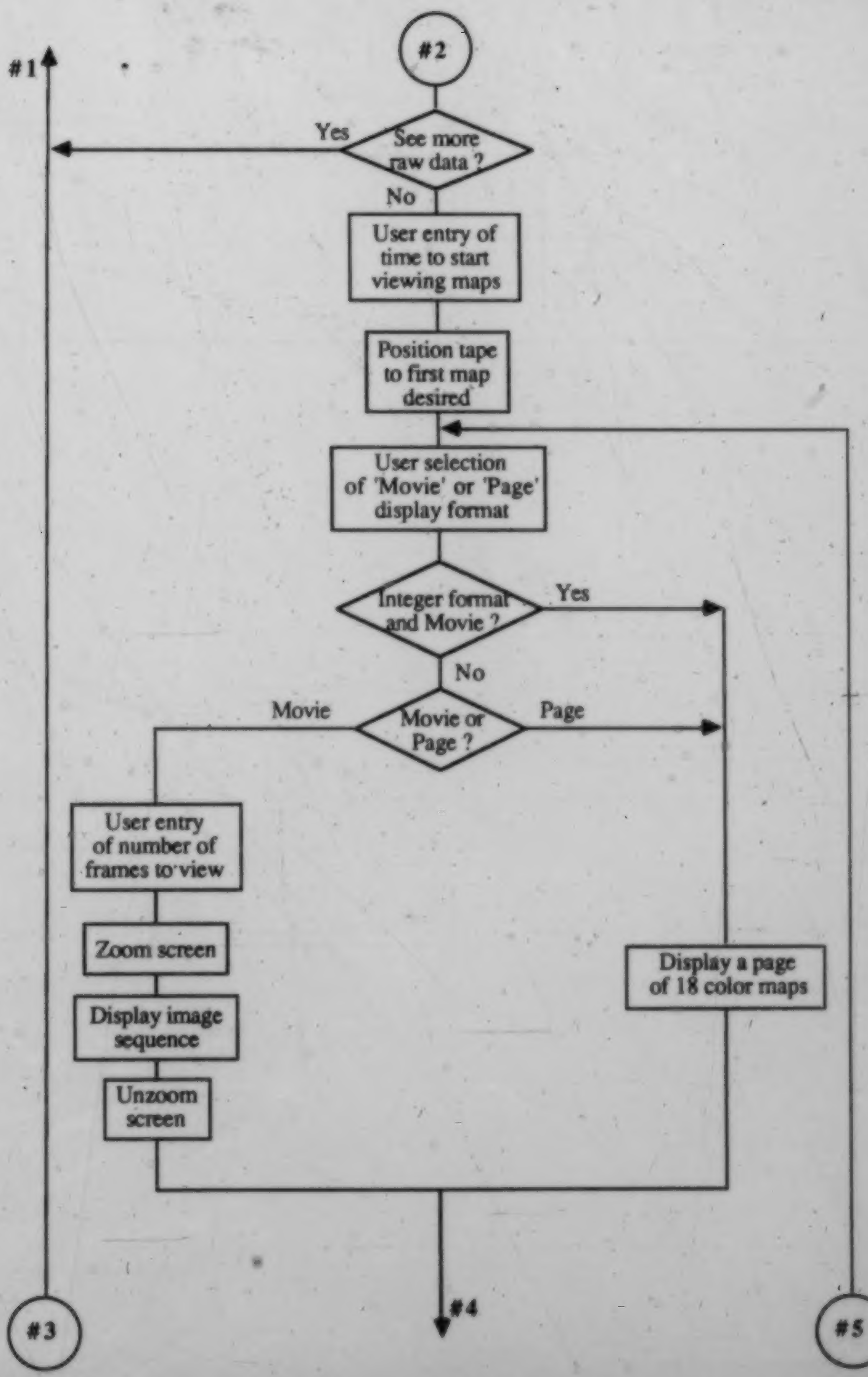
Get header file

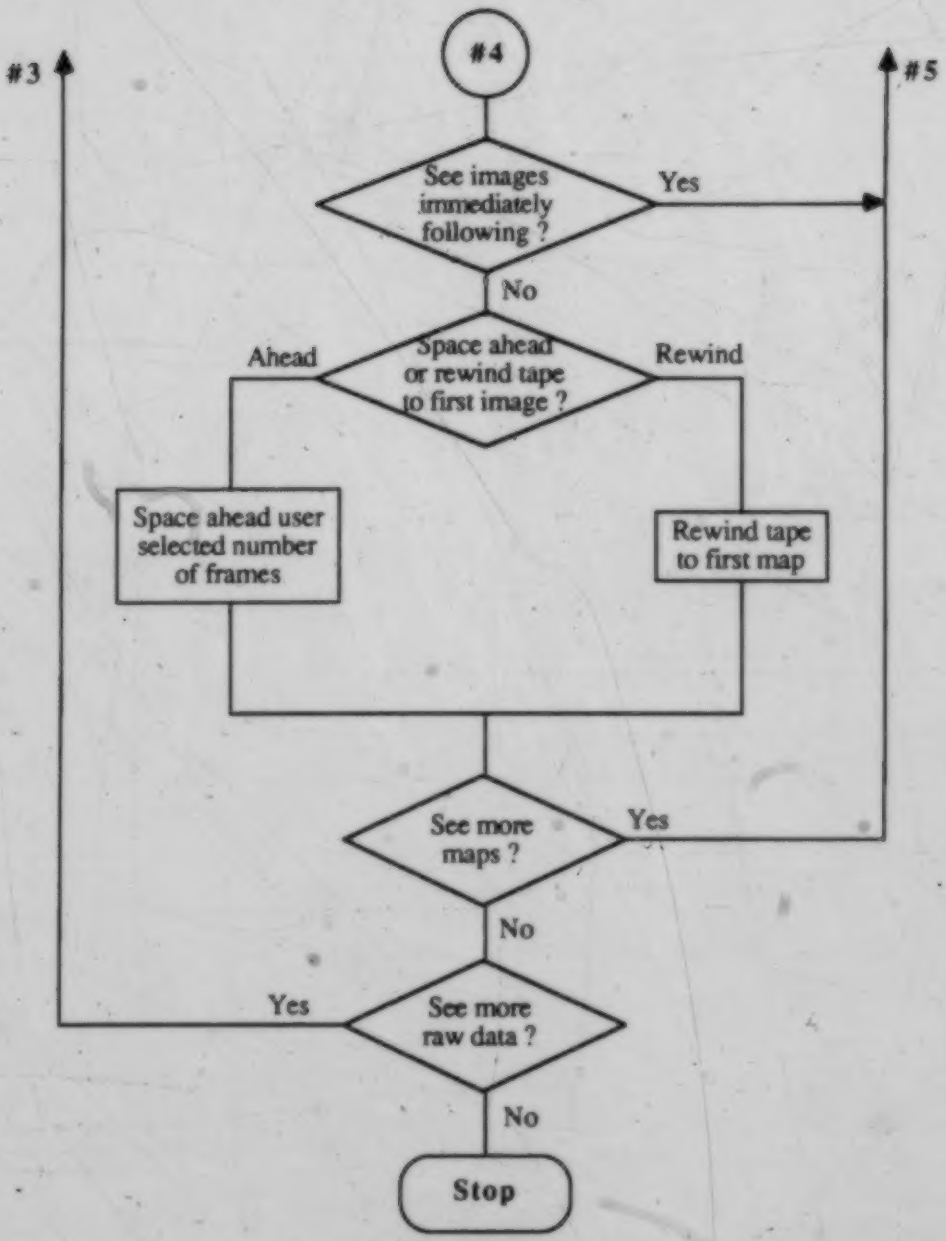
This section of code gets the header information describing the mapping options and file descriptions from the streaming tape. The user is given the opportunity to review this information and to see whether the right tape has been inserted into the machine.

Get raw data

After reviewing the header information, the user is given the opportunity to review the raw data file on the tape. Since there are 2040 sample sets on tape, and only 640 pixels across the color graphics display, the raw data is divided into 4 @ 32 channel pages of 510 samples each. For visual clarity, the users are allowed to see only 8 channels of data at a time from whichever page they select.


Start
Get header file
Display header information
Correct tape ?
No
Stop
Yes
Get raw data
Draw electrode chart
User entry of 8 channels for raw data display
User entry of raw data page
Page 1 ?
No
Yes
Page 2 ?
No
Yes
Page 3 ?
No
Yes
Page 4 ?
No
Yes
Display 8 selected channels
Display 8 selected channels
Display 8 selected channels
Display 8 selected channels
#1
#2


#2
#1
Yes
See more raw data ?
No
User entry of time to start viewing maps
Position tape to first map desired
User selection of 'Movie' or 'Page' display format
Integer format and Movie ?
Yes
No
Movie
Movie or Page ?
Page
User entry of number of frames to view
Zoom screen
Display image sequence
Unzoom screen
Display a page of 18 color maps
#3
#4
#5


#4
#3
#5
See images immediately following ?
Yes
No
Space ahead or rewind tape to first image ?
Ahead
Rewind
Space ahead user selected number of frames
Rewind tape to first map
See more maps ?
Yes
No
See more raw data ?
Yes
No
Stop

RDTAPE flowchart

Figure 3.10

After reviewing a page of raw data, the user has the option of continuing to review raw data, or moving on to display topographic maps. Each raw data display has a horizontal time scale across the bottom to give the user a time reference into the topographic map file on the tape.

Position tape

When the user is ready to go on to review topographic maps, RDTAPE issues a prompt for the user to input a time reference at which to position the tape in the map file. Tape position is calculated directly from the time, at which point the tape drive spaces ahead an appropriate number of maps.

Movie or Page format ?

The user has the option of viewing topographic maps in a page of 18 sequential maps as described earlier, or as a movie, displaying one map after another in an enlarged format. In the movie format, each map can be read sequentially from the tape and displayed in turn, rather than reading all required maps into memory and having them displayed. This is done because of the prohibitive memory requirements of the latter approach; however, due to the slow retrieval speed of the tape drive, maps can only be reviewed at about 1 frame per second.

Viewing more maps

After viewing the sequence of maps specified above, the user has the option of continuing to view maps in sequence, spacing ahead a specified number of frames, rewinding the tape to the first map, reviewing the raw data, or quitting.

3.3 Interpolation methods

It is useful to make comparisons between various methods of interpolation, to evaluate their suitability to the problem at hand. Accordingly, the frequency response characteristics of one dimensional linear and cubic spline interpolations are examined. Next, the behavior of the cubic spline is examined under the 'free' and 'clamped' boundary value assumptions. Finally, the behavior of the bi-cubic spline is examined under the two classes of boundary value assumption, as well as for various assumptions for corner values necessitated by this particular application.

3.3.1 Frequency response characteristics

It was indicated in chapter 2 that interpolation can be considered a low pass filtering process. This will be the basis for comparing the frequency response of linear and cubic spline interpolation. It is useful to first note that both linear and cubic spline interpolation are linear processes. That is, the coefficients which describe the interpolating polynomials are linear functions of the data values obtained from regular sampling. Specifically, it can be seen that for linear interpolation, the general expression for the interpolating polynomials is

$$S(x) = \sum_{k=1}^{N-1} M_k(x-x_k) + y_k \qquad \text{where } M_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{\Delta y_k}{\Delta x_k}$$

and (x_k, y_k), [$k = 1,2,3,\ldots,N$] are the sampled data points

When the data is regularly sampled,

$$\Delta x_k = \Delta x_{k+1} = C \ , \quad \text{for all } k \ \ (\text{ C a constant })$$

The M_k, which are simply the slopes of the linear interpolating polynomials, are clearly linear functions of the values y_k. For cubic spline interpolation on the other hand, the general expression for the interpolating polynomials is

$$S(x) = \sum_{k=1}^{N-1} A_k(x-x_k)^3 + B_k(x-x_k)^2 + C_k(x-x_k) + D_k$$

Examination of the set of relations (I) and (II) and equations 2.6 and 2.9 in section 2.1.2.2 shows that the coefficients A_k, B_k, C_k and D_k of the cubic interpolating polynomials, are linear functions of the values y_k, when the data are sampled regularly.

Since both linear and cubic spline interpolation are linear processes, the principle of superposition can be applied; that is, each data point can be interpolated independently of each other to produce a set of point spread functions associated with each sample location. These point spread functions form the basis of their respective interpolation methods. Thus, to compare the frequency response of linear and cubic spline interpolation, the frequency response of their respective point spread functions can be examined.

The point spread function of an ideal low pass filter is of the form (sin x)/x. This function is sometimes referred to as an infinite order interpolant, owing to the power series expansion for sin x. The point spread function for linear interpolation is the chateau function illustrated in figure 3.11. The point spread function for cubic spline interpolation has a form more closely resembling (sin x)/x and is illustrated in figure 3.12 for both the 'free' and the 'clamped' boundary conditions. The chateau function of figure 3.11 can be viewed as a first order approximation to (sin x)/x while figure 3.12 can be viewed as a third order approximation to (sin x)/x. In the case of cubic spline interpolation, the shape of the interpolating function is clearly affected by the choice of boundary condition. However, as the left and right boundaries approach infinity, the 'free' and 'clamped' splines become equal.

The Fourier transform of the chateau function is well known [4], to be of the form

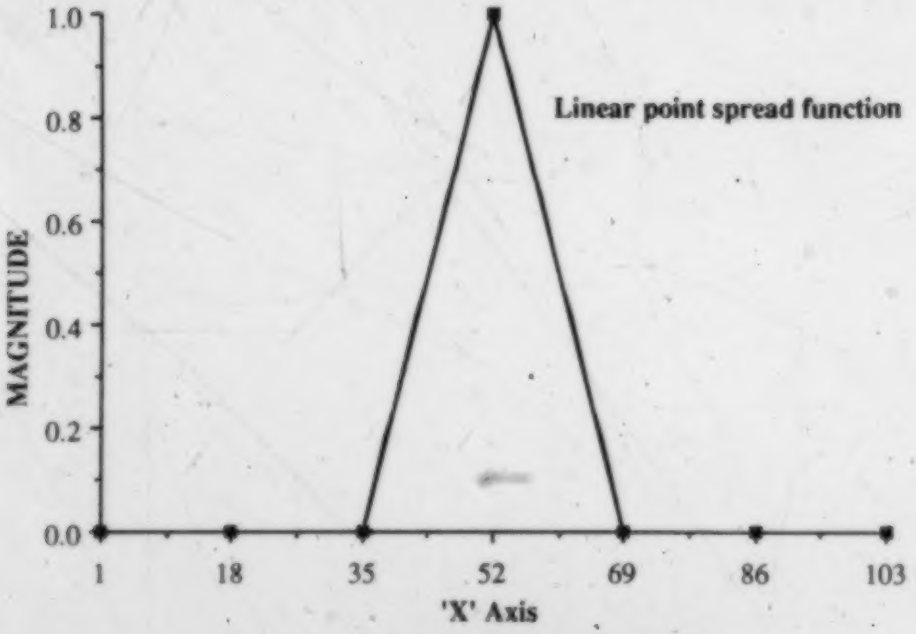
$$S(\omega) = K \frac{\sin^2 x}{x^2}$$

An analytical expression for the frequency response of the cubic spline point spread function can be obtained from a straight forward, although messy, application of the Fourier transform,

$$S(\omega) = \int_{-\infty}^{\infty} \sum_{k=1}^{N-1} \left(A_k (x - x_k)^3 + B_k (x - x_k)^2 + C_k (x - x_k) + D_k \right) e^{-j\omega x} \, dx$$

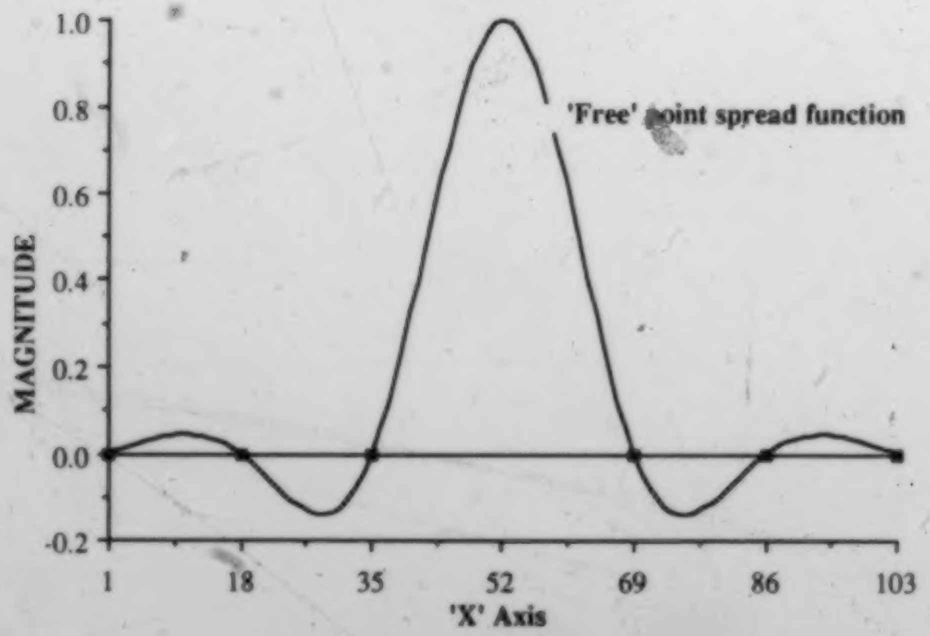
For the purposes of this report, a 1024 point FFT will be used to illustrate the differences between the linear and cubic spline interpolants.

In the discussion which follows, it will be considered that a continuous one dimensional bandlimited image is to be sampled at a sampling rate of 10. Further, an interpolated image is desired which would resemble the continuous image as if it was sampled at a sampling rate of 100. For comparison purposes, an ideal low-pass filter with a cut-off frequency of 5 (the folding frequency) would be sufficient to ideally interpolate the image. Further, the boundaries will be considered to be far apart, producing negligible boundary effect for the cubic spline interpolator. Figure 3.13 illustrates the magnitude response for the linear, cubic and ideal interpolators.



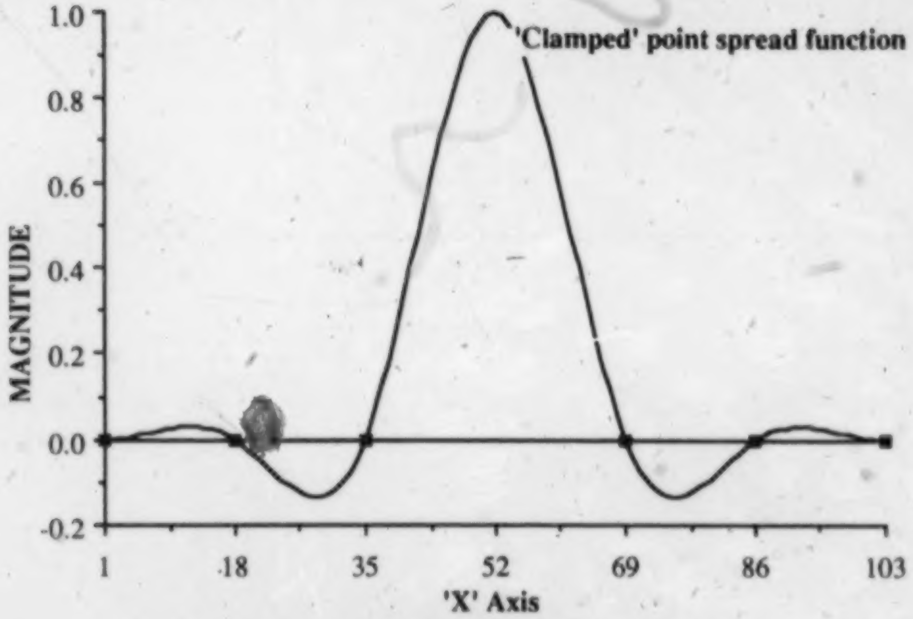
Point spread function (chateau function)
for linear interpolation

Figure 3.11



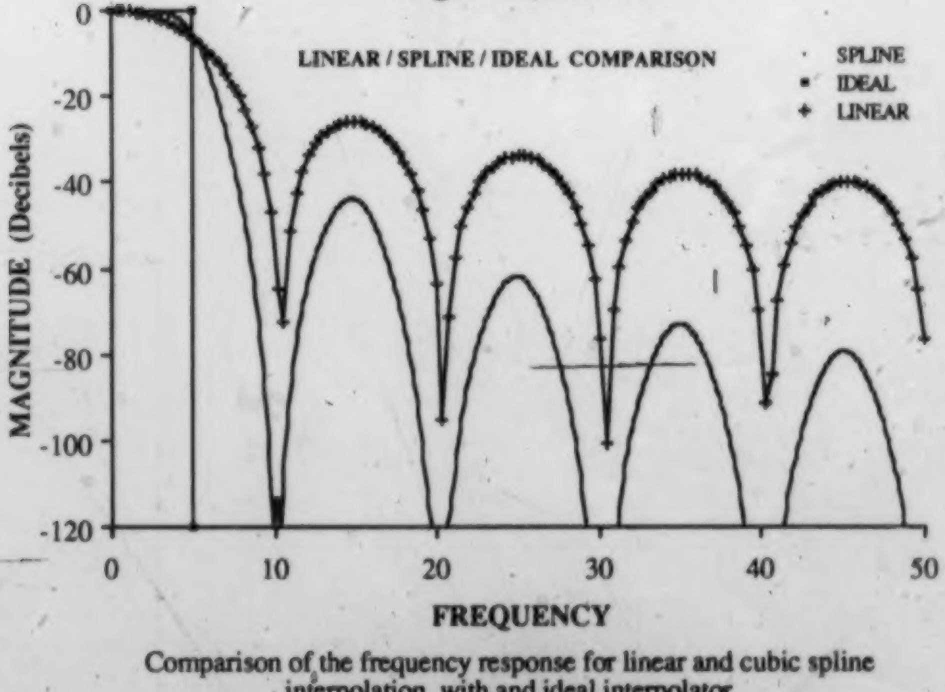
Point spread function for cubic spline interpolation
using the 'free' boundary condition

Figure 3.12 a



Point spread function for cubic splineinterpolation using the 'clamped'boundary condition

Figure 3.12 b



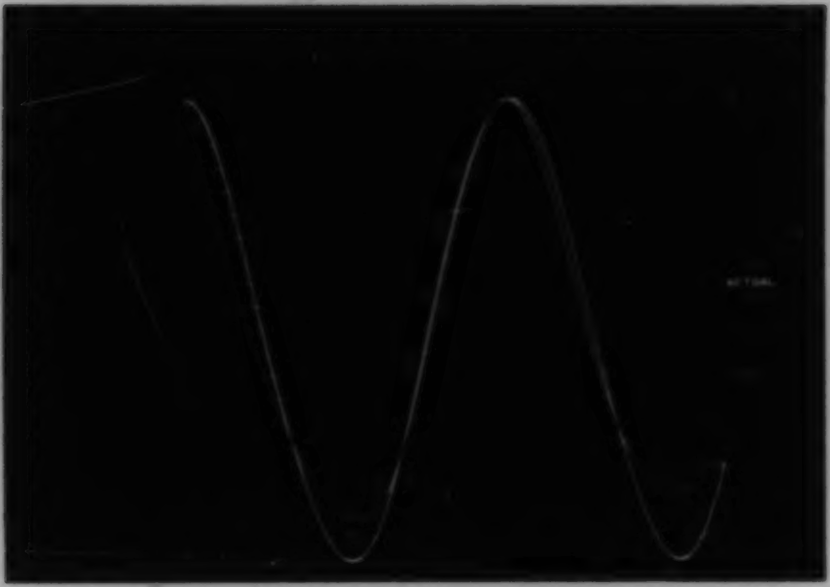
Comparison of the frequency response for linear and cubic spline interpolation, with and ideal interpolator

Figure 3.13

3.3.2 One dimensional spline behavior

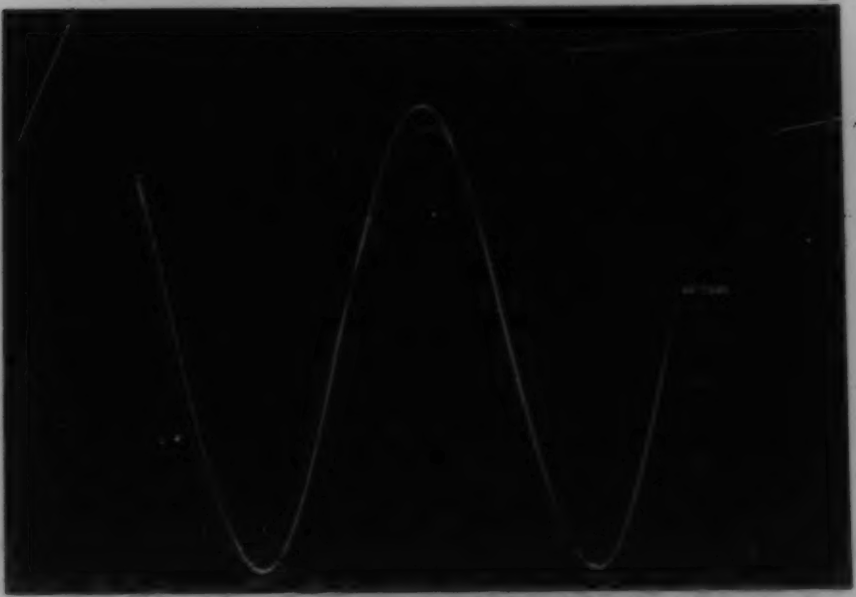
To test the 1-D spline algorithms which are core to the bi-cubic spline algorithm under the 'free' boundary condition or the 'clamped' boundary condition (taken as $d^2f_1/dx^2 = d^2f_n/dx^2 = 0$ and $df_1/dx = df_n/dx = 0$ respectively) program SPLINE was created. SPLINE employed the Spath [39] algorithms with subroutines CUBICI and CUBI2, which are described in appendix B. The approach taken by SPLINE was to plot an analytic function which could be the summation of up to 5 sinusoidal functions of selectable amplitude, frequency and phase, and periodically sample it 5 times over the plotting interval. This sampling rate implies a Nyquist rate of 2.5 cycles. The sample set obtained by this procedure was then interpolated under the 2 boundary conditions mentioned above and superimposed on the plot of the actual function sampled. This gave a good comparison between the performance of the 'clamped' and 'free' cubic splines in a variety of situations and provided an good illustration of how far cubic spline interpolation can be pushed relative to the Nyquist criterion.

Figure 3.14 is a photographic illustration of what SPLINE did in one particular situation. Samples used in the spline subroutines were taken at each vertical line in the background grid. The signal plotted is a single 1.6 cycle sinusoid with zero phase shift in 3.14 (a) and 73 degrees phase shift in 3.14 (b). The functions can be expressed as $\cos(1.6x + \phi)$ where $0 \leq x \leq 2\pi$ and ϕ represents the phase shift in degrees. This form of expression will be used in all future analytical representations of actual sinusoidal functions which are sampled and interpolated by SPLINE.



Actual function $f(x) = \cos 1.6x$

Figures 3.14 (a)



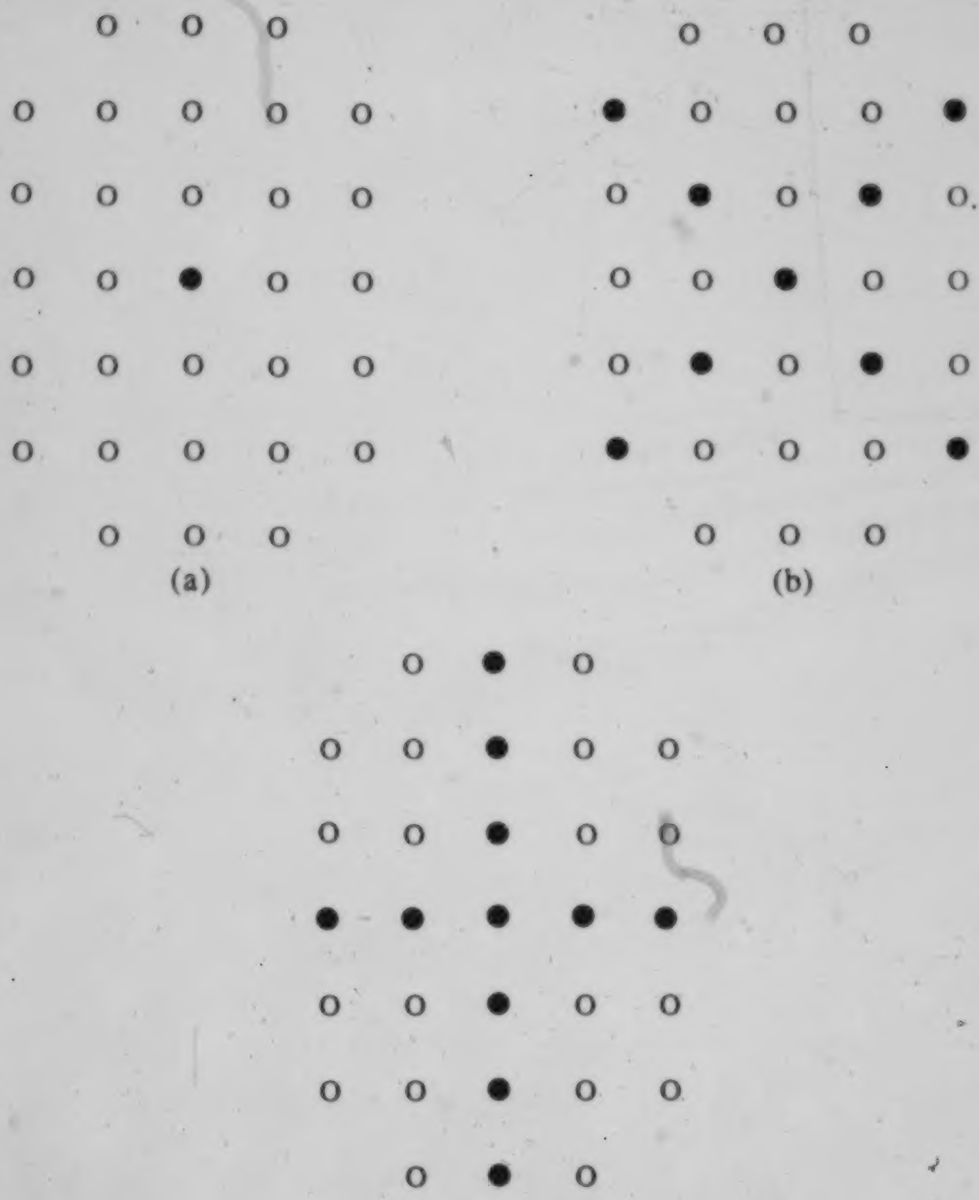
Actual function $f(x) = \cos(1.6x+73^{\circ})$

Figures 3.14 (b)

3.3.2 Two dimensional spline behavior

In addition to the tests conducted on the cubic spline, the author felt it would be useful to perform tests on the bi-cubic spline to gain a 2 dimensional perspective on the performance of splines under the 'free' and 'clamped' boundary condition. In addition, an issue peculiar to the 2 dimensional problem, in this application, and which has already been mentioned was to examine the effects of various assumptions made about corner values in the sample grid. In the topographic mapping application described here, there were no electrode positions which corresponded to these positions.

The tests that were done on the bi-cubic spline consisted of doing a voltage map interpolation, according to the equations of section 3.2.1.3, on a variety of 2 dimensional sample patterns. These patterns can be considered as 2 dimensional arrays of equally weighted dirac delta functions on a zero background. The test patterns used are illustrated in figures 3.15 and 3.16. The patterns in figure 3.15 were created to compare the bi-cubic spline under the 'free' and the 'clamped' boundary conditions. Figure 3.15 (a) is a single dirac delta function on a zero background, which when interpolated will illustrate the impulse response of the interpolating spline (as discussed in chapter 2, interpolation is ~~basically a~~ low-pass filtering operation). Figure 3.15 (b) and (c) are just 2 additional patterns for comparing 'free' and 'clamped' spline interpolations.



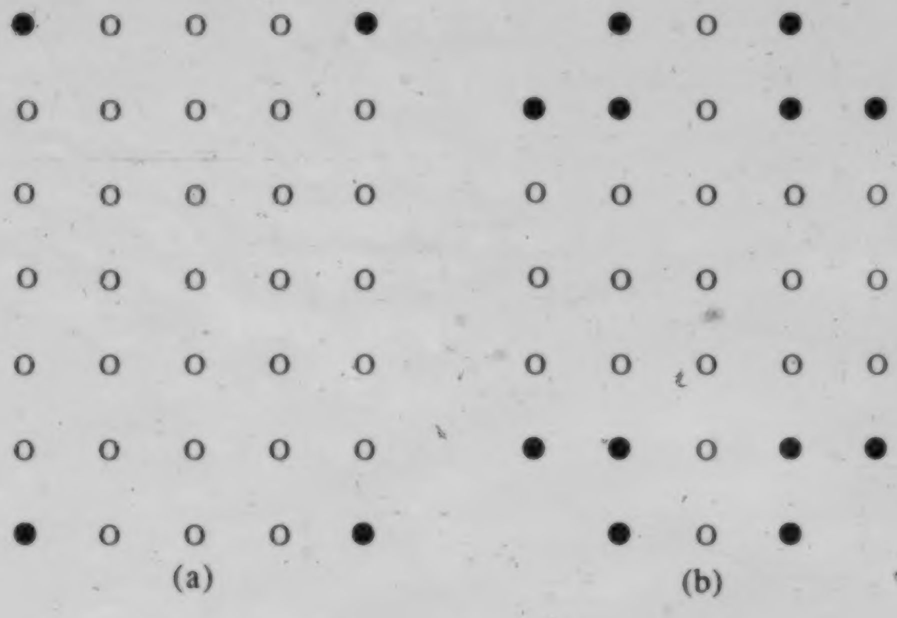
(a)

(b)

(c)

Bicubic spline test patterns
Figure 3.15

To assess the effect of the corner electrodes on the sampled image, the test arrays of figure 3.16 were created. In 3.16 (a) the corner positions were turned on and interpolation was performed under both of the aforementioned boundary conditions. In 3.16 (b), the 3 nearest neighbors for each corner electrode were turned on. At this point, 'free' and 'clamped' voltage interpolations were done (again according to the voltage equations of section 3.2.1.3), first with the a corner value taken as a bi-linear average of the 3 nearest neighbors, and then with the corner value set to zero.



(a) (b)

Corner value test patterns
Figure 3.16

As a final test of the behavior of the bi-cubic spline, a section of actual data representing 18 contiguous image frames was mapped according to the current equations of section 3.2.1.3. First, 'clamped' splines were used and the interpolated results compared for 'zeroed' and 'averaged' corners. Finally, 'free' splines were used and the interpolated results compared for 'zeroed' and 'averaged' corners. The results of all tests are presented in chapter 4 and discussed in chapter 5.

Chapter 4
PRESENTATION OF TEST RESULTS

Chapter 3 presented the mapping system hardware and software of this report, and methods used to test the spline interpolation techniques employed therein. This chapter presents the results obtained from the testing procedures of section 3.3. Discussion of these results is left to chapter 5.

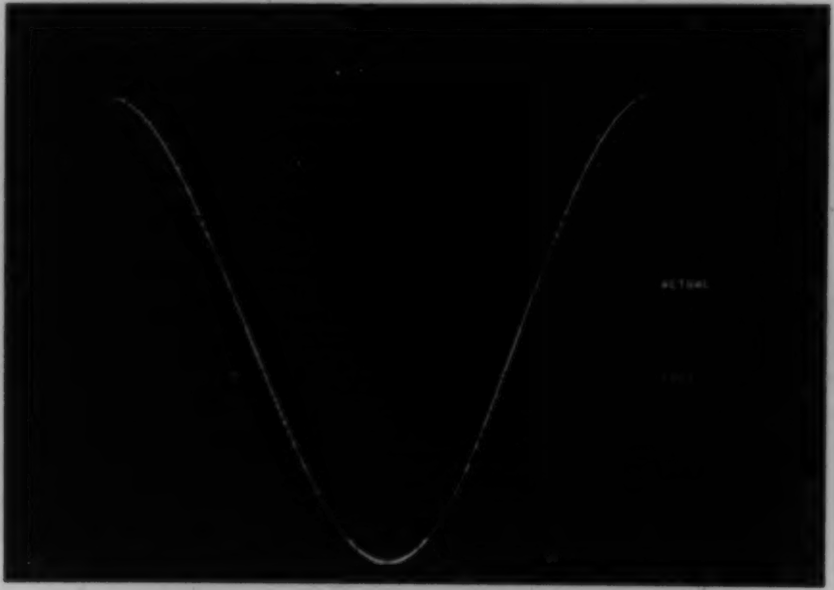
4.1 1-D spline test results

Figure 4.1 is a series of 14 photographs taken of the testing described in section 3.3.1. Here, the 'free' and 'clamped' splines were compared for several sinusoidal functions. Since the Nyquist frequency is 2.5, (spatial frequency information will be specified without the interval^{-1} unit) the actual function sampled and interpolated was varied between a frequency of 1.0 and 2.0. Because a single sinusoid with zero phase shift could be expected to provide an advantage for the 'clamped' boundary condition, 2 different phase shifts of 0 degrees and 45 degrees were used to provide a balanced comparison between the 'free' and 'clamped' splines.

Figures 4.1 (a) and (b) present an actual function which is a single sinusoid with a frequency of 1, which is plotted in white. In Figure 4.1 (a), the actual function is given a phase shift of 0 while in 4.1 (b) a phase shift of 45 degrees is introduced. In these photographs, as with all the photographs of figure 4.1, the actual function is plotted in white, and sampled at each vertical line on the background grid. For the sample set so obtained, the 'clamped' spline is computed, and plotted in red while the 'free' spline is plotted in green. For ease of comparison, the spline interpolations are superimposed on the actual underlying function. In figures 4.1 (c) and (d) the frequency of the actual underlying signal is increased to 1.5. The 0 degree phase shift is shown in (c) while the 45 degree phase shift is shown in (d). In figures 4.1 (e) and (f) the frequency of the actual function is incremented to 1.8 while in figures 4.1 (g) and (h) frequency is incremented again to 2.0. It can be seen, that the quality of the interpolation is quite good up to a frequency of 1.5. However, departures from the actual curve are more apparent as frequency is increased beyond this point and it is not immediately apparent whether the departures are due to inflexibility of the cubic spline or boundary effects.

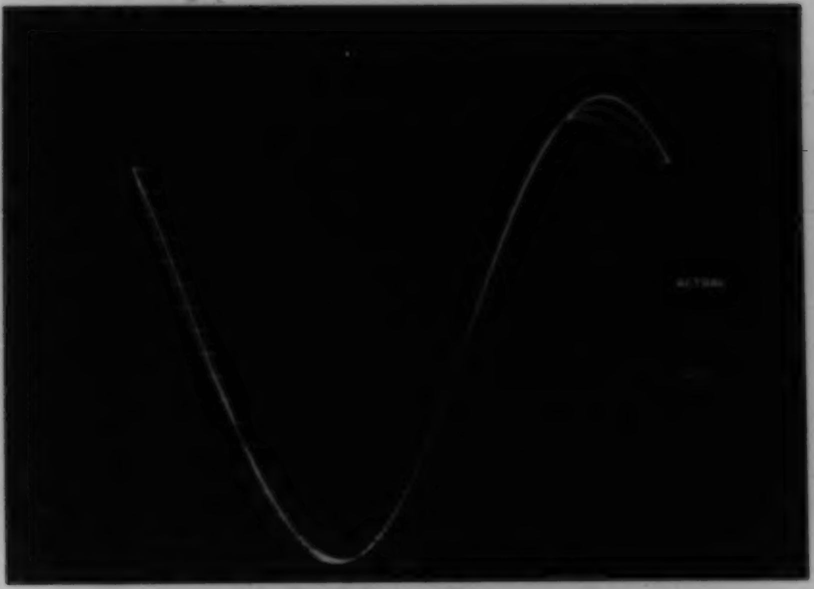
The author felt that at this point program SPLINE would be able to provide an excellent illustration of the concept of aliasing which is well known in sampling theory. In sampling theory it is known that when a signal is sampled, its frequency domain representation is repeated about integer multiples of the sampling frequency. Thus if a 1 Hz signal is sampled at 5 Hz, the frequency domain description of the sampled signal should repeat at 5 - 1 = 4 Hz and 5 +1 = 6 Hz. It is perhaps less often discussed however how phase shifts in the sampled signal affect the frequency domain replication process. With these ideas in mind, figures 4.1 (i) through (l) were created. The reader is invited to compare these 4 photographs with figures 4.1 (a) and (b).

Finally, it was felt that it would be useful to present a cubic spline interpolation of a more complex signal. Accordingly, figures 4.1 (m) and (n) were created. In (m), a composite of 3 sinusoids is presented while in (n) a composite of 5 sinusoids which press the limits of interpolation is presented. In each case, the analytic description of the signal sampled is presented in the caption of each photograph. Further discussion of these results is left to the discussion of chapter 5.



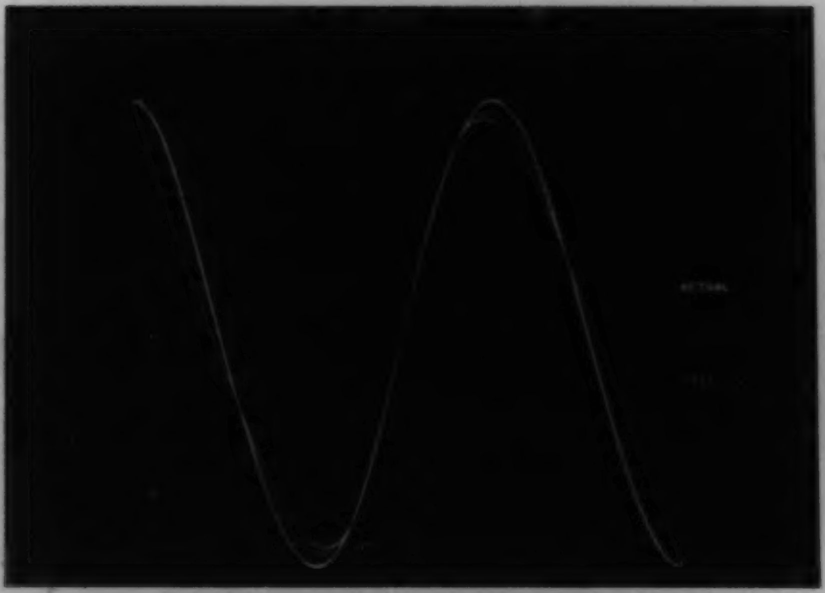
Actual function $f(x) = \cos x$

Figure 4.1 (a)



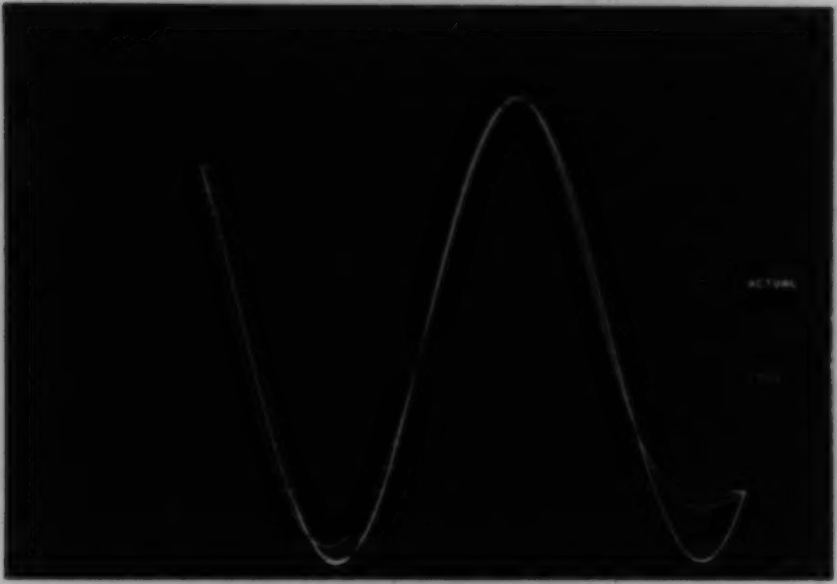
Actual function $f(x) = \cos(x+45^\circ)$

Figure 4.1 (b)



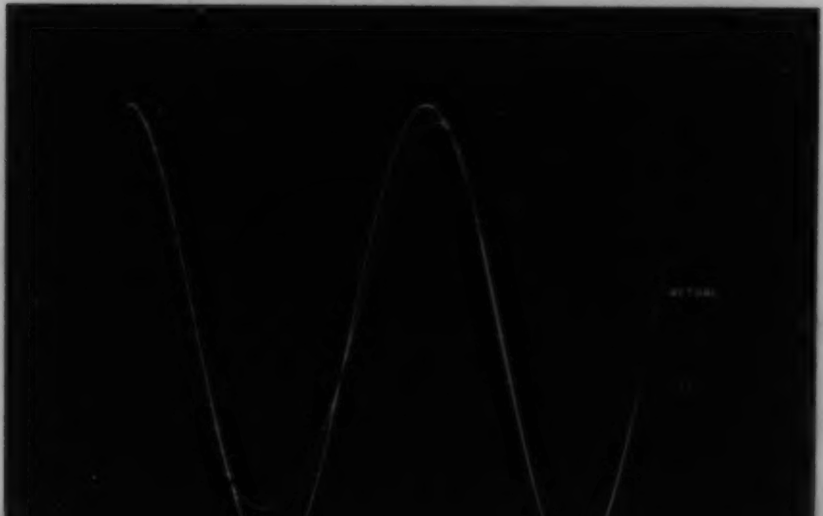
Actual function $f(x) = \cos 1.5x$

Figure 4.1 (c)



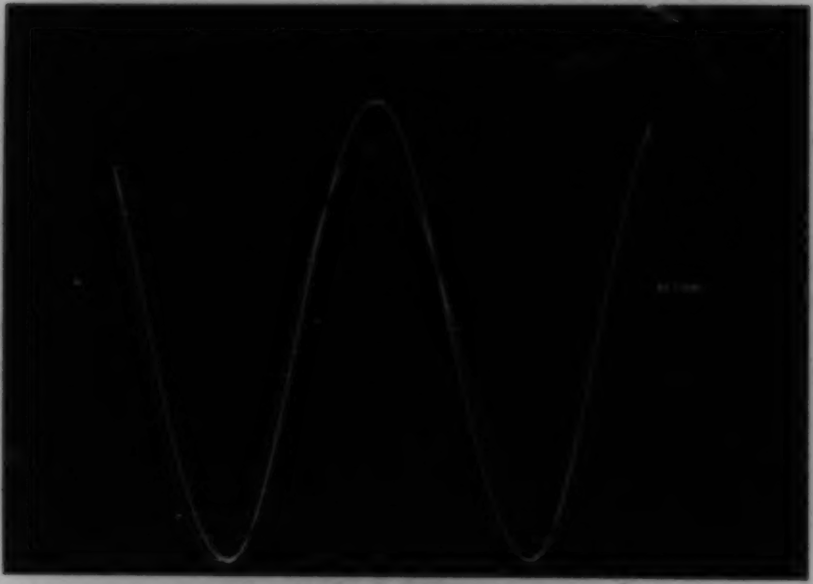
Actual function $f(x) = \cos(1.5x+45^\circ)$

Figure 4.1 (d)



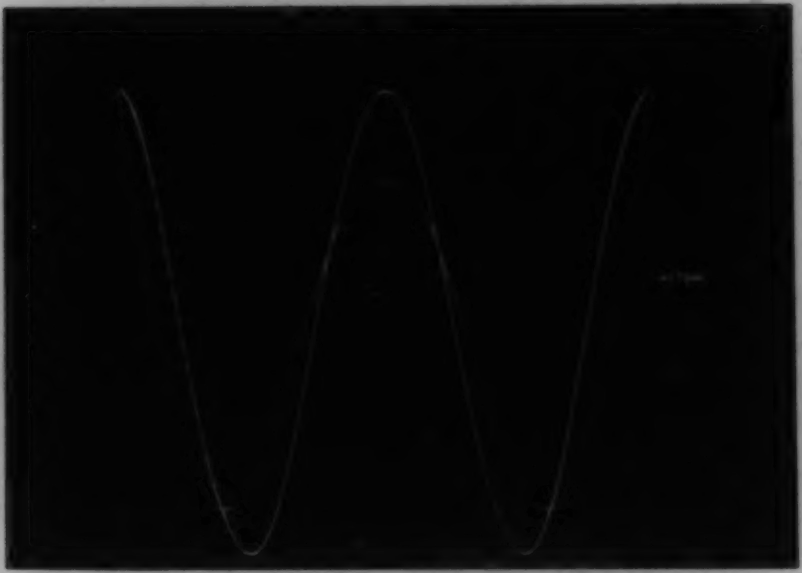
Actual function $f(x) = \cos 1.8x$

Figure 4.1 (e)



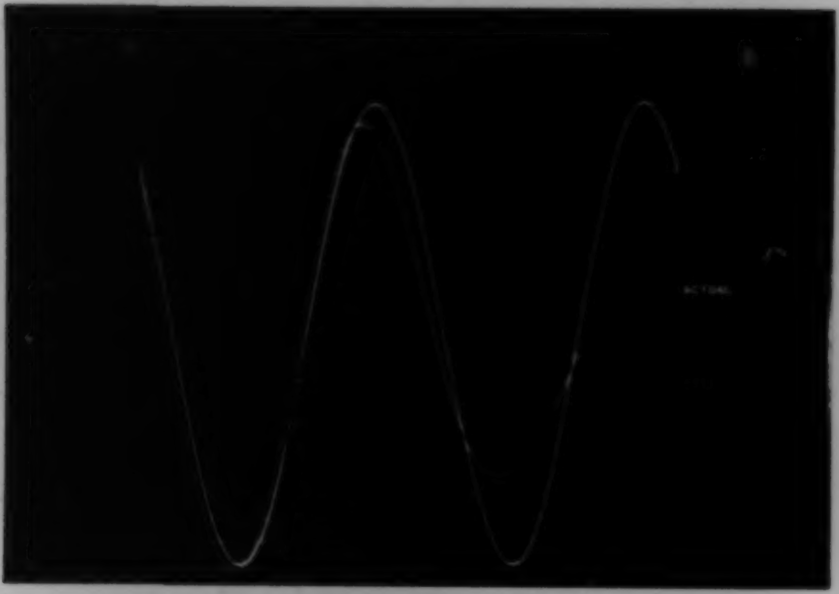
Actual function $f(x) = \cos(1.8x+45°)$

Figure 4.1 (f)



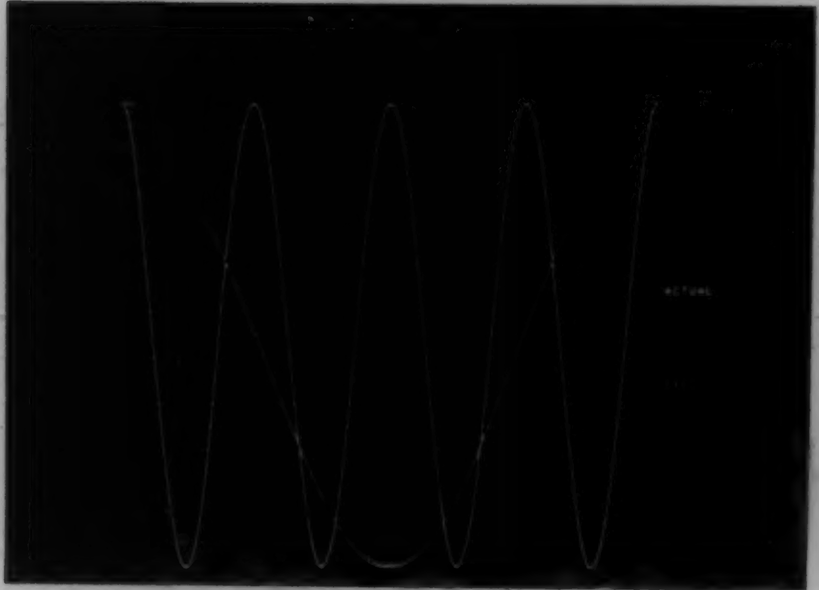
Actual function $f(x) = \cos 2x$

Figure 4.1 (g)



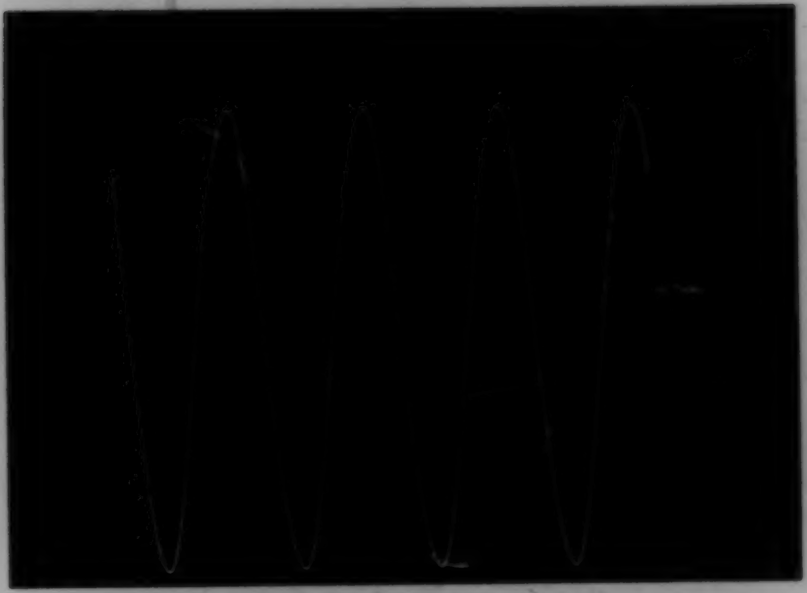
Actual function $f(x) = \cos(2+45^{\circ})$

Figure 4.1 (h)



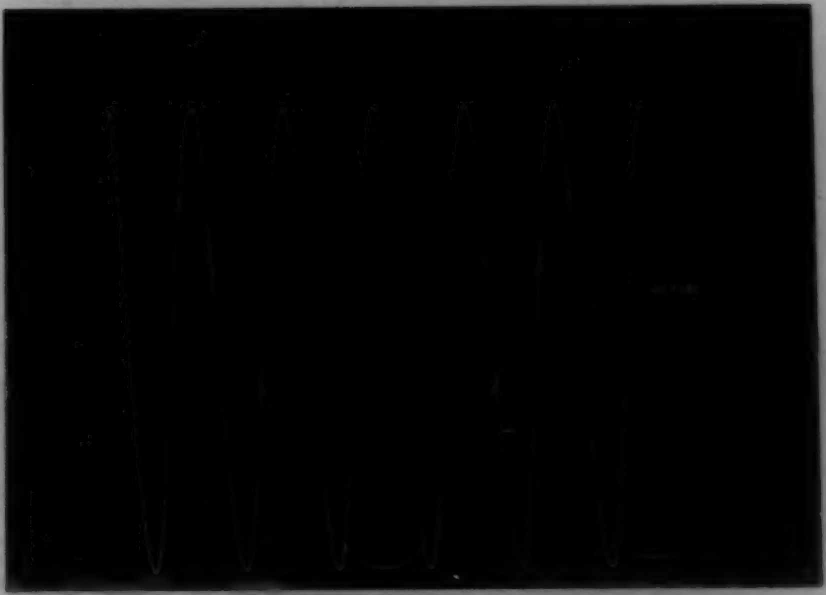
Actual function $f(x) = \cos 4x$

Figure 4.1 (i)



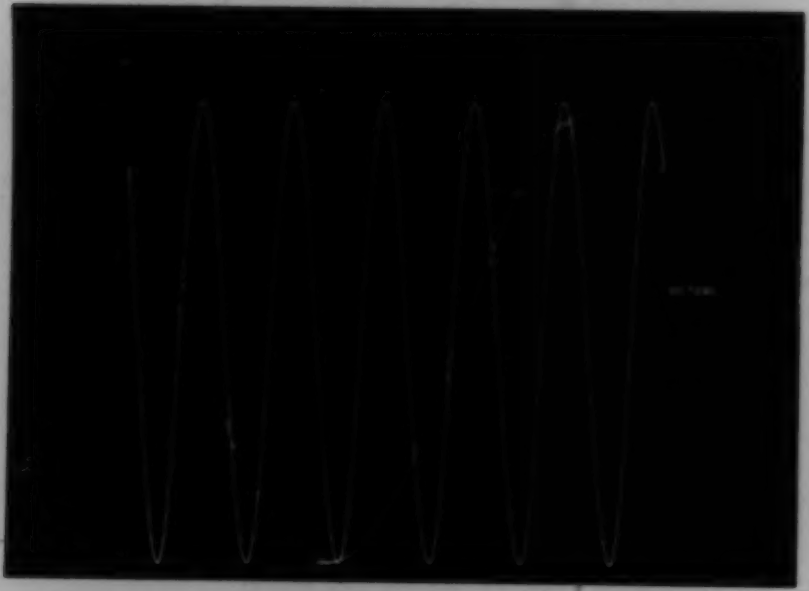
Actual function $f(x) = \cos(4x+45^\circ)$

Figure 4.1 (j)



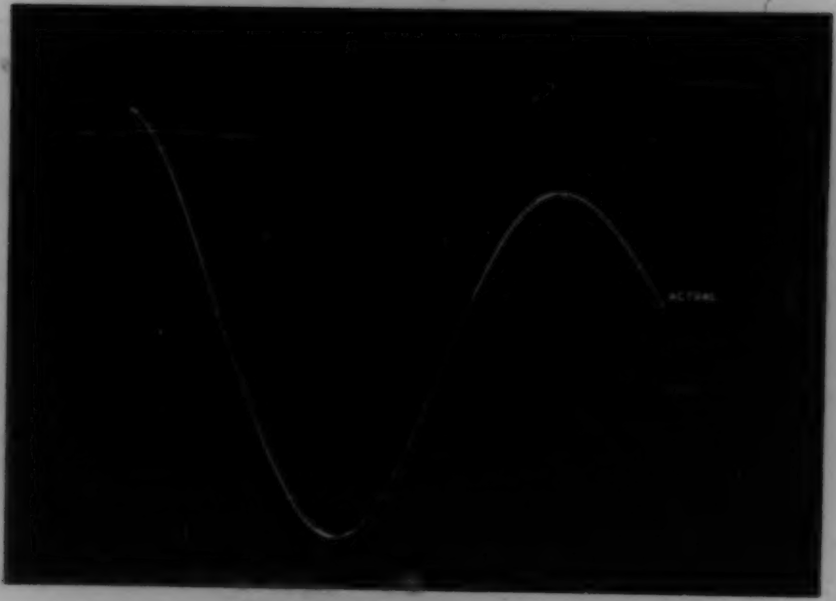
Actual function $f(x) = \cos 6x$

Figure 4.1(k)



Actual function $f(x) = \cos(6x+45^\circ)$

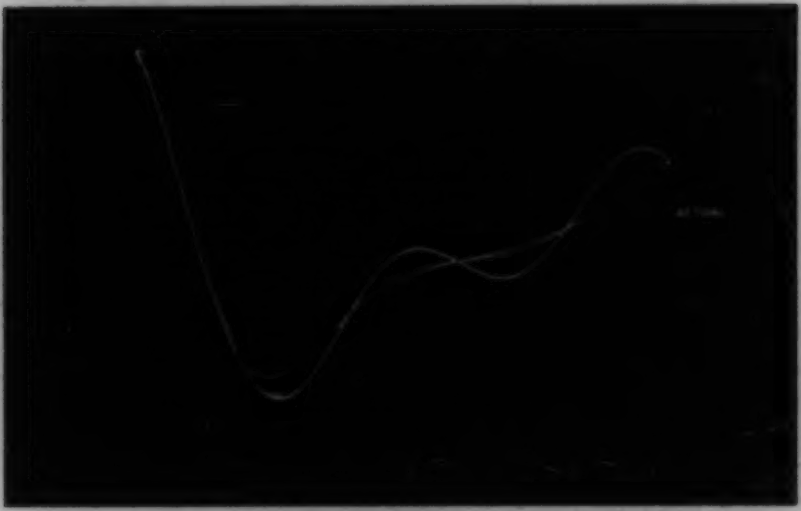
Figure 4.1(l)



Actual function

$f(x) = \cos x + 1.2\cos(1.2x+10^\circ) + 0.8\cos(1.4x+20^\circ)$

Figure 4.1(m)



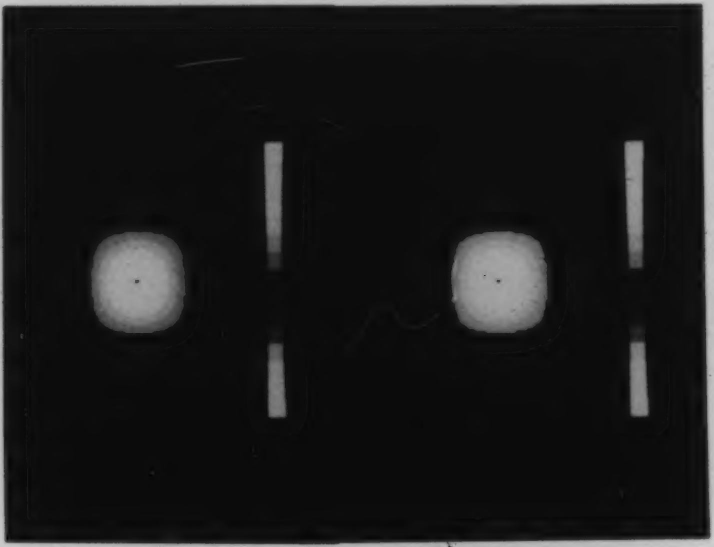
Actual function
$f(x) = \cos x + 1.2\cos(1.2x+10^\circ) + 0.8\cos(1.4x+20^\circ)$
$+ \cos(1.8x+30^\circ) + 1.2\cos(2x+40^\circ)$

Figure 4.1 (n)

4.2 2-D spline test results

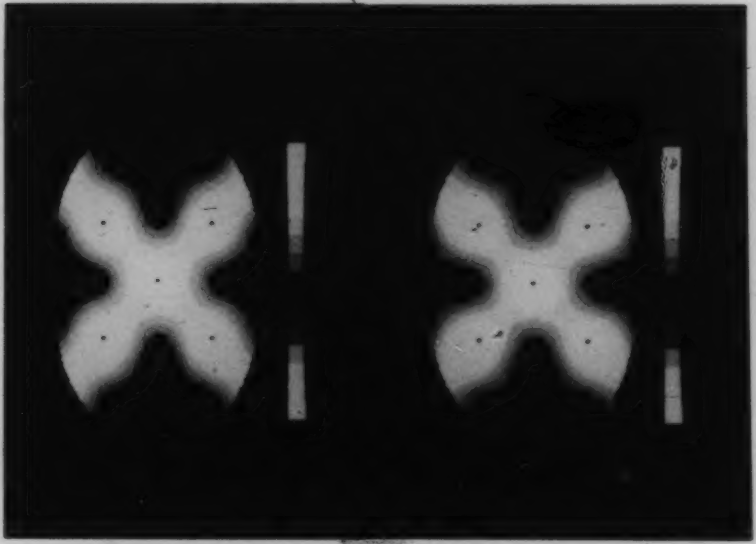
Figures 4.2 to 4.5 are a series of photographs which present the bi-cubic spline tests mentioned in section 3.3.2. The figure 4.2 photographs are voltage map interpolations of the test patterns presented in figure 3.15. These patterns were created in the same spirit as the cubic spline tests presented in the section above; that is to compare the effects of the 'free' and 'clamped' boundary conditions on the topographic mapping process. Figures 4.2 (a), (b) and (c) compare voltage maps of the test patterns in figure 3.12 (a), (b) and (c) respectively, under the 'clamped' boundary condition ($\partial S_{1j}/\partial x = \partial S_{nj}/\partial x = 0$, $\partial S_{1i}/\partial y = \partial S_{mi}/\partial y = 0$ and $\partial^2 S_{11}/\partial x\partial y = \partial^2 S_{n1}/\partial x\partial y = \partial^2 S_{1m}/\partial x\partial y = \partial^2 S_{nm}/\partial x\partial y = 0$) and the 'free' boundary condition ($\partial^2 S_{1j}/\partial x^2 = \partial^2 S_{nj}/\partial x^2 = 0$, $\partial^2 S_{1i}/\partial y^2 = \partial^2 S_{mi}/\partial y^2 = 0$ and $\partial^2 S_{11}/\partial x\partial y = \partial^2 S_{n1}/\partial x\partial y = \partial^2 S_{1m}/\partial x\partial y = \partial^2 S_{nm}/\partial x\partial y = 0$). In figures 4.2 to 4.4 the 'clamped' spline interpolation appears on the left, while the 'free' spline interpolation appears on the right.

From figure 4.2 (a), it can be seen that the 'clamped' boundary condition compresses the sidelobes of the point spread function for bi-cubic splines more than does the 'free' boundary condition. This is particularly evident in the horizontal direction of the image as compared with the vertical direction. Examination of figure 4.2 (b) reveals that the 'clamped' boundary condition has effects at the boundary of the ellipse that are not present in the 'free' spline map. The same effects are not as apparant in the interpolations of figure 4.2 (c).



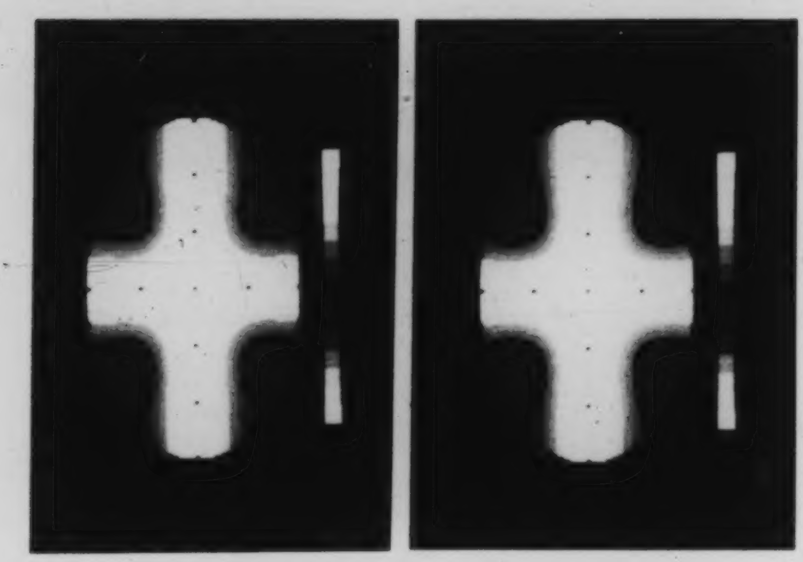
Interpolated voltage map for test pattern 3.15 (a)
(clamped and free boundary conditions)

Figure 4.2 (a)



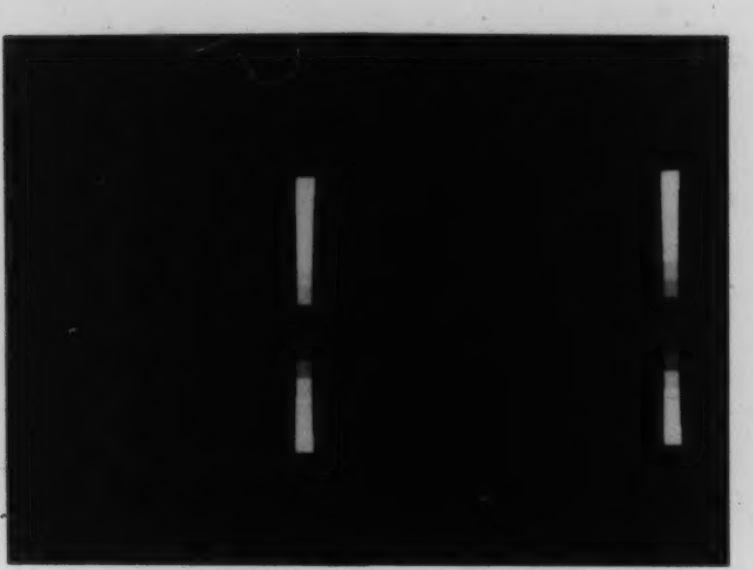
Interpolated voltage map for test pattern 3.15(b)
clamped and free boundary conditions)

Figure 4.2 (b)



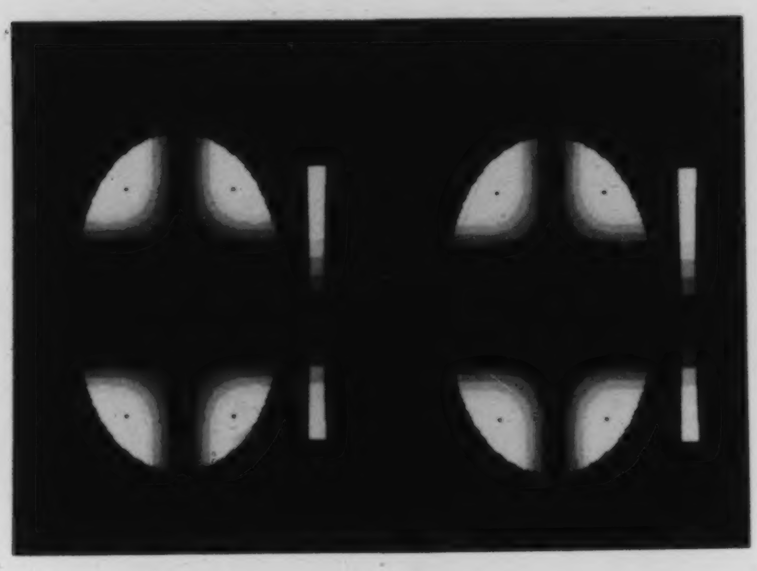
Interpolated voltage map for test pattern 3.15 (c)
(clamped and free boundary conditions)
Figure 4.2 (c)

To test the effect of corner value assumptions, the test patterns in figure 3.16 were created. The topographic mapping system provides the user with an opportunity to choose either a zero corner value, or a value which is a bi-linear weighted average of the 3 electrode positions neighboring the corner position. It was desired, that these assumptions be tested for both 'clamped' and 'free' spline interpolations. The photographs in figure 4.3 are voltage map interpolations of the test pattern in figure 3.16 (a), which is intended to test the effect of the corner electrodes themselves. The photographs in figure 4.4 are voltage map interpolations of the test pattern in figure 3.16 (b).
In figure 4.3 the left image is a voltage map interpolation under the 'clamped' boundary condition while the right image is a voltage map interpolation under the 'free' boundary condition, for the test pattern of figure 3.16 (a). It is apparent in figure 4.3 that the effect of the corner electrode is not present within the elliptical window used to present a topographic map, when the 'free' boundary condition is used, while the corner electrode has a noticable effect under the 'clamped' boundary condition.



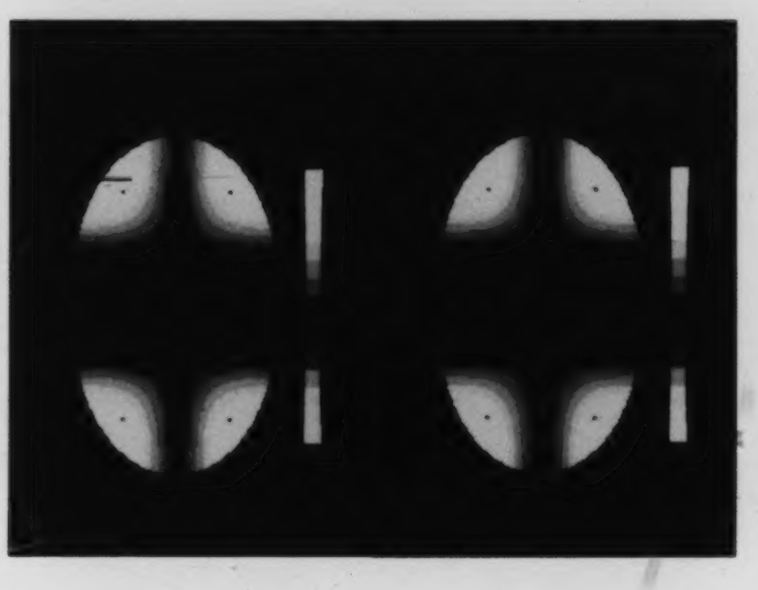
Interpolated voltage map for test pattern 3.16 (a)
(clamped and free boundary conditions)
Figure 4.3

In the interpolations of figure 4.4, the effect of choosing a bi-linear average can be compared with the choice of a zero corner value. Furthermore, the effect of these choices is presented under both the 'clamped' and the 'free' boundary conditions. The maps of figure 4.4 are again voltage map interpolations, this time of the test pattern in figure 3.16(b). From these maps, the choice for the corner value seems to have less effect upon the interpolations than the choice of boundary condition.



Interpolated voltage map for test pattern 3.16 (b)
Averaged and zeroed corners (clamped boundary condition)

Figure 4.4 (a)

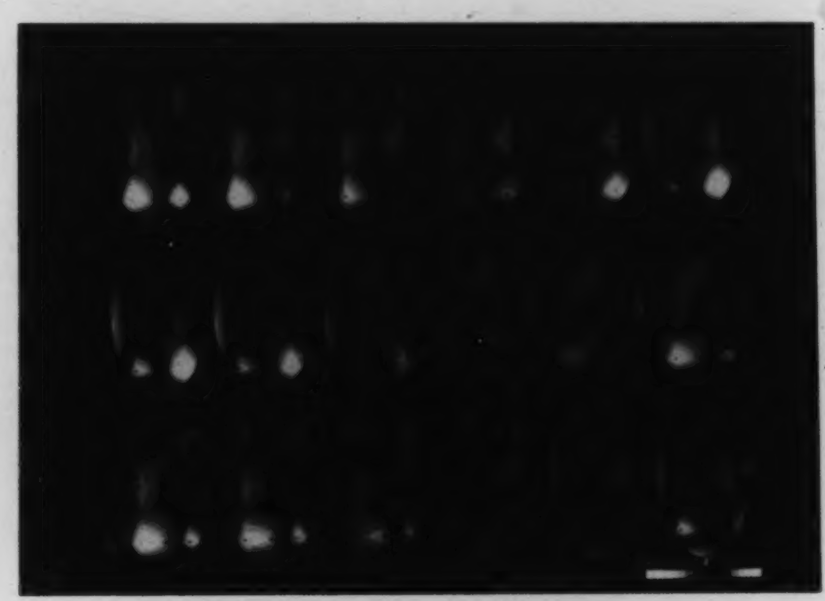


Interpolated voltage map for test pattern 3.16 (b)
Averaged and zeroed corners (free boundary condition)

Figure 4.4 (b)

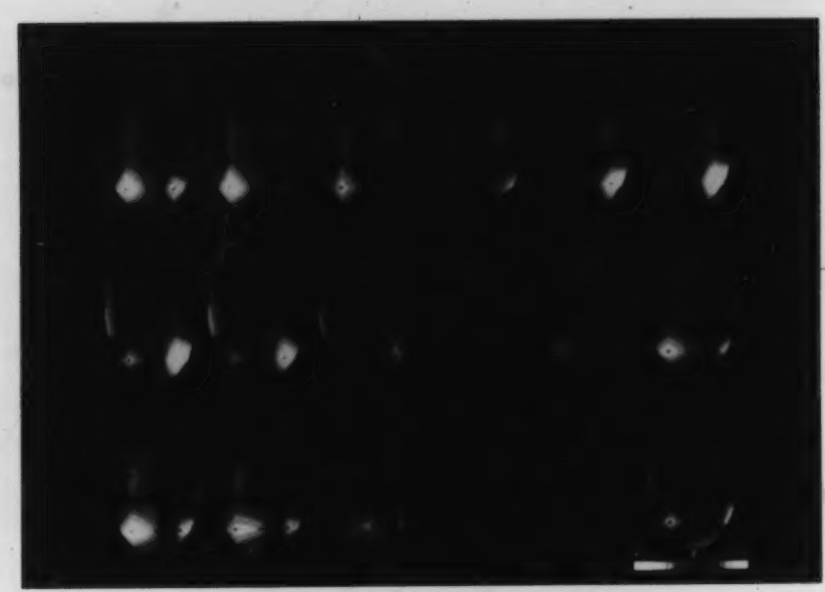
As a final test of the effects of boundary condition choices and corner value choices, on the topographic mapping process, source current maps of actual EEG data are presented in figure 4.5. A single 18 frame section of actual EEG data was temporally filtered between 8 and 13 Hz., interpolated frame by frame according to the current equations of section 3.2.1.3, and displayed as a single page of 18 maps. Figures 4.5 (a) and (b) illustrate the effect of corner value selection under the 'clamped' boundary condition while figures 4.5 (c) and (d) illustrate the effect of corner value selection under the 'free' boundary condition.

It can be seen that there were very minor differences between the 2 pages of maps made under the 'clamped' boundary condition or between the 2 pages of maps made under the 'free' boundary condition: that is, the selection of average or zeroed corner values seemed to have only a minor effect. However, it is clear that there were considerable differences introduced into the mapping process, owing to the selection of boundary values. There appeared to be structural similarities in all the maps generated in this test, but all the 'clamped' spline maps seemed to exhibit a boundary effect not present in the 'free' spline maps, and which seems more pronounced in the horizontal direction than in the vertical direction.



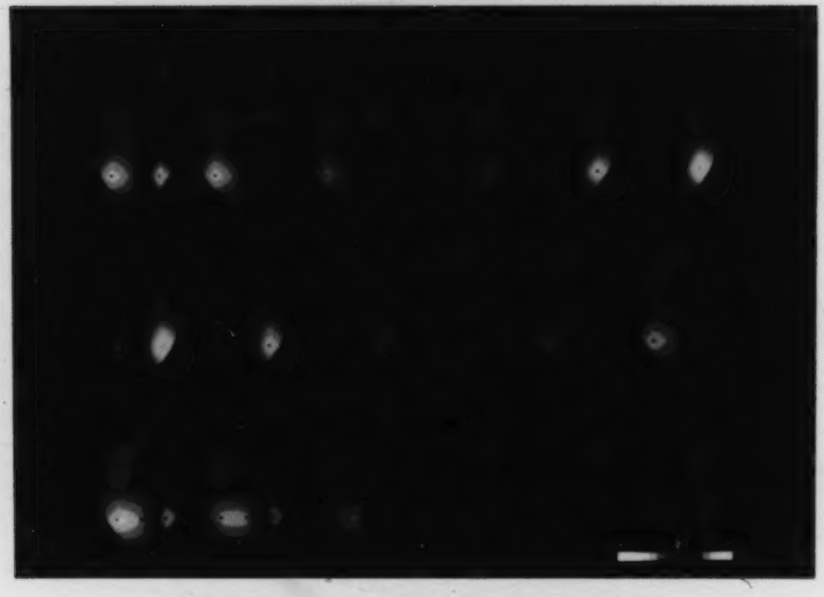
Current maps of actual data (Alpha rhythm)
Averaged corners (clamped boundary condition)

Figure 4.5 (a)



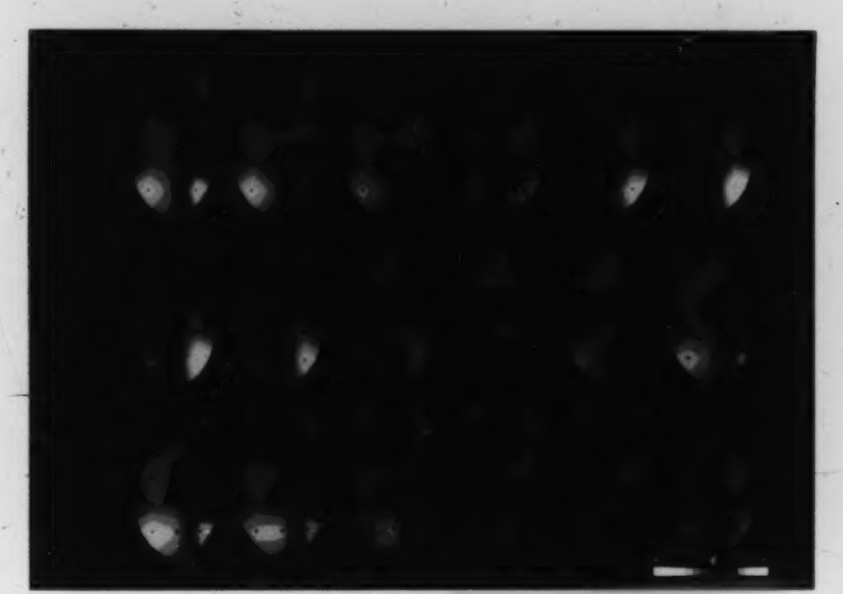
Current maps of actual data (Alpha rhythm)
Zeroed corners (clamped boundary condition)

Figure 4.5 (b)



Current maps of actual data (Alpha rhythm)
Averaged corners (free boundary condition)

Figure 4.5 (c)



Current maps of actual data (Alpha rhythm)
Zeroed corners (free boundary condition)

Figure 4.5 (d)

Chapter 5

DISCUSSION AND CONCLUSIONS

In the 60 odd years since its discovery, the EEG has advanced primarily as a clinical tool in medicine. As mentioned previously however, the manner in which the EEG is interpreted has changed little. It is well understood that a problematic aspect of the utility of the EEG is that the conventional polygraphic presentation confronts the electroencephalographer with a large amount of multi-dimensional information which is difficult to decompose consistently purely on a visual basis. Topographic mapping has emerged as a promising method to advance the clinical utility of the EEG and it is the advancement of this area of research to which this thesis was addressed. Specifically, the issues of electrical reference, spatial sampling and presentation of the EEG have been addressed and will now be discussed.

5.1 Electrical reference

The electrical reference question is one which has continued to confound the routine application of the EEG. This is probably the most significant issue to which this thesis addressed itself and radial current mapping was presented as an effective way to deal with this issue. By applying the Laplacian operator to the electric potential surface, maps of radial current flow can be obtained. These maps are unaffected by the variability of the electrical reference and map scalp electrical currents, which better localize generators in the cortex. Source current mapping is widely accepted as being preferential to voltage mapping [14-20,29,36] and this work is among the first to apply the technique to topographic mapping of actual EEG.

A significant advantage that this thesis has had, is that higher order interpolation techniques have been employed (bi-cubic splines) in the map generation process. As will be discussed in following sections, the sampling density of the spatial EEG is likely to increase rather than decrease, which will in turn bring about an increasing level of performance on the part of the bi-cubic spline in topographic mapping applications. Admittedly, bi-cubic spline interpolation is significantly more complex than bi-linear methods employed to date, but their use permits the analytical application of the Laplacian to a smooth interpolated surface. Analytical application of the Laplacian to smooth surfaces is in turn a significant improvement over application of digital approximations of the

Laplacian to first order interpolations. The power of modern computer technology provides the needed answer to the problem of increased complexity.

5.2 Spatial Sampling

Gevins [22] has suggested that as many as 100 electrodes may be necessary to adequately sample the spatial EEG, based upon evidence he has found that the point-spread function is as little as 1 cm in some cases. Nunez [35] has pointed out that in other situations, cerebral generators in the brain are sometimes comprised of more extensive dipole sheets, indicating a smaller requirement for spatial sampling density. Currently unpublished work within the Department of Applied Sciences in Medicine at the University of Alberta, suggests that if the temporal frequency is below 15 Hz, a 25 electrode array is sufficient. It may be that the frequency correlate is relevant to Gevins observation, but further discussion of this is beyond the scope of this thesis. Of course, this frequency limitation restricts application of the mapping process to beta band activity, which extends from about 13 Hz to 20 Hz or higher. Duffy's work has been based upon the more conventional 16 to 20 electrode array [10]. Duffy has also gone on to perform numerous clinical studies of various forms of cerebral dysfunction. Lehmann [30,31] on the other hand uses a 48 electrode array to sample the spatial EEG. The 31 electrode array used in this work represents an improved sampling density relative to much of the topographic mapping work done to date.

Certainly, the spectral extent of the spatial EEG merits further study and this will imply increasing the sampling density of the spatial EEG. Thus, future topographic mapping systems will incorporate 60 or more electrodes. For instance, to provide a small increase in sampling density by going to a 7 x 10 electrode array would require 67 electrodes. Work is currently in progress, within the department of Applied Sciences in Medicine, to develop a data acquisition system which will accommodate up to 128 channels and will no doubt help to shed more light on the question of the spatial EEG. It is worth mentioning however, that to routinely apply 100 or so electrodes will require an improved electrode placement technology.

5.3 Topographic presentation

5.3.1 Signal conditioning

A central aspect of the topographic mapping process is the ~~desired~~ temporal signal conditioning applied to the EEG prior to the interpolation process. That is, it may be desired to bandpass a selected rhythmicity such as alpha waves, and map this activity to portray the spatial characteristics of the rhythm. This thesis has employed digital filter design techniques which are readily implemented into the mapping process once EEG data has been digitized and stored. The bandpass filter design process then becomes a choice between IIR and FIR filter designs. The theoretical aspects of these filter design techniques were described in chapter 2.

The choice between an IIR or FIR type digital filter becomes a choice between the linear phase characteristics / large coefficient counts of FIR filters and the non-linear / small coefficient counts of IIR filters. In this thesis, the author has preferred the zero phase characteristics of FIR designs to eliminate phase distortion when studying the EEG. As will be discussed, the increased computational complexity of FIR designs is small in proportion to implementing bi-cubic splines.

An area of EEG research which has been popular, but which is considered outside the scope of this thesis is that of temporal artifact identification and / or rejection due to muscle movement or interfering signals. If the rejection of these artifact signals results in significant fragmentation of the temporal continuity of EEG recordings, then edge effects resulting from the application of FIR filters becomes more pronounced, since mapping should not be performed when these edge effects are present.

Another point which manifested itself during the work, was the importance of fully utilizing the dynamic range of analog / digital conversion equipment while recording the EEG. It was found that the signal strength varied widely between subjects, so while a front end amplifier gain setting which was acceptable for one subject, caused signal clipping in other situations or was insufficient in others. The results would then be adversely affected by either too much quantization noise, or distortion resulting from signal clipping. Thus considerable adjustment of amplifier gain settings in the data acquisition system should be provided to achieve an optimum balance between either of these two extremes.

5.3.2 Bi-cubic spline interpolation

By far the largest proportion of effort in this thesis concerns the application, testing and evaluation of bi-cubic splines for the purposes of topographic mapping. Indeed, sections 2.1.2.2, 2.1.2.3, 3.3.1, 3.3.2 and all of chapter 4 have been dedicated to this area. This section is thus devoted to the evaluation of test results presented in chapter 4.

5.3.2.1 Cubic spline evaluation

Figures 4.1 (a) through (i) presented a series of cubic spline interpolations of sampled sinusoidal signals. The sampling frequency in these photographs is 5 and the actual functions range in frequency from 1 to 2. It will be noted in these photographs, particularly in figure 4.1 (a), that the 'clamped' spline interpolation appears to perform better than the 'free' spline interpolation. However, it must be remembered that in the case of figure 4.1 (a), as with 4.1 (c), (e) and (g), the clamping boundary value assumption of $dS_1/dx = 0$ happens to be exactly correct, whereas when a phase shift is introduced to the actual signal, this assumption is not correct. Figures 4.1 (b), (d), (f) and (h) illustrate this point quite clearly, and in fact illustrate that in the general case when a non-zero phase shift is present in the sampled signal the 'free' spline can perform somewhat better.

The key to understanding whether a spline with a 'clamping' boundary value or a 'free' spline should be used lies in the a priori knowledge of the behavior of the sampled signal at the boundaries. The 'clamped' boundary condition in fact assumes that such knowledge is available. The 'free' boundary condition, specifically $d^2S_1/dx^2 = d^2S_n/dx^2 = 0$, makes no such assumption and in fact, in the absence of a priori slope information, trains the interpolating polynomials not to do any bending at the boundaries. This lack of bending at the boundaries, on the part of the 'free' spline is readily visible in figures 4.1 (a) through (h).

It is also apparent from figures 4.1 (a) through (h) that as the frequency of the sampled signal approaches the Nyquist limit imposed by the sampling rate, the quality of the cubic spline interpolations decreases expectedly. In fact, the departure of the cubic spline interpolations is quite significant at a signal frequency of 2 and is even noticeable at a signal frequency of 1.5 as illustrated in figures 4.1 (h) and (d) respectively at the right hand boundary. These 2 photographs illustrate that as signal frequency approaches the Nyquist limit, interpolation errors due to erroneous boundary value assumptions, propagate further inward from the boundary. However, if the boundary value assumption can approach reality, cubic spline interpolation performs quite well, even at 2.0/2.5 = 0.8 or 80% of the

Nyquist frequency. In the more general case a complex signal where boundary value slope information is unavailable, figures 4.1 (m) and (n) illustrate that the cubic splines can perhaps be pushed as high as 1.8/2.5 = .72 or 72% of the Nyquist frequency. Furthermore, under these conditions, the 'free' boundary condition is the most general and most reliable assumption to make.

In summary, it can be seen from the above that cubic spline interpolation is restricted when the behavior of the sampled signal is unknown at the boundaries of the sampling window. More sophisticated techniques such as Maximum Entropy [27] make the most general assumptions possible about the behavior of the sampled signal outside the sampling window, given the constraints of the signal's behavior inside the sampling window. However, such techniques represent an additional order of complexity which must be dealt with if they are to be effectively implemented. The 'free' boundary condition for cubic spline interpolation functions solely on the strength of the knowledge available about the sampled signal within the sampling window. Thus, given the evidence presented in figure 4.1, the 'free' boundary condition should be used for cubic spline interpolations of signals such as the spatial EEG where behavior of the potential surface is unknown at the boundaries of the sampling grid.

5.3.2.2 Bi-cubic spline evaluation

Further tests of cubic splines were performed in 2 dimensions, and these results are presented in section 4.2. The test results presented in figure 4.2 are intended to compare the performance of the 'free' and 'clamped' boundary conditions in 2 dimensions. As such, figures 4.2 (a), (b), (d) and (e) most effectively demonstrate the differences between the effects of these boundary conditions. It was pointed out in section 4.2 that the 'clamped' spline maps were more compressed and exhibited more ringing at the boundary, particularly in the horizontal direction, than were the 'free' spline maps. The differences in the horizontal and vertical directions are due to the fact that while the sampling frequency is the same in both directions, there are fewer samples in the horizontal direction so that boundary value errors discussed in section 5.3.2.1 above, propagate further toward the center of the maps.

The corner value tests presented in the photographs of figure 4.3 indicate that selection of a corner value presents the most noticeable effects on voltage maps, under the 'clamped' boundary condition. In fact, the effect is not even detected in the map of figure 4.3 (b). In figure 4.4, the effect of choosing a zero corner value is compared to that of choosing a bi-linear average for the corner value. The differences between these assumptions are noticeable under both the 'free' and the 'clamped' boundary condition as is noticeable from a comparison of figures 4.4 (a) and (b) or from a comparison of figures 4.4 (c) and (d). Again however, the effect of the boundary condition appears to be dominant.

The final test of corner value selection is presented in figure 4.5. Here corner value assumptions were compared under the 'free' and 'clamped' spline boundary conditions for an actual EEG data file. The data file exhibited a high degree of alpha activity and was filtered between 8 and 13 Hz. prior to mapping. The maps generated were current maps rather than voltage maps and in each test case presented in figure 4.5 they readily illustrate the predominance of alpha activity in the occipital region, consistent with known attributes of the alpha rhythm in humans. Comparing the 'free' and 'clamped' spline maps, a consistency of structure can be seen in the maps, but feature compression and boundary ringing is present in every 'clamped' boundary condition map. In fact, the differences between 'free' and 'clamped' spline far out weigh differences in the maps due to the corner value assumption.

From the above evidence, the 'free' (sometimes called natural) spline interpolation method was preferred by the author, for generating topographic maps. Further, the calculation of bi-linearly averaged corner values is not warranted given the evidence presented above and in view of the additional computation they require. As with the discussion in section 5.3.2.1 above concerning boundary value assumptions; that is, given the absence of accurate or reliable estimates for unknown values, the author feels it is better to make the most general assumption possible.

5.3.3 Displaying topographic maps

Viewing topographic maps in a movie type fashion has been quite popular in topographic mapping work done to date [6,8-10,34], and it has been suggested during this work, that a topographic mapping system would ideally be capable of processing and displaying maps in real time. To examine the real time idea however, the temporal frequencies present in EEG, the response characteristics of the human visual system and the sampling rate used for the EEG must be considered. It was mentioned earlier, that humans begin to perceive continuous motion at about 10 images / sec; but humans also cannot distinguish between images presented at rates much beyond 30 images / sec. Alpha band activity in the EEG is typically between 8 and 13 Hz., while Beta band activity is between about 13 and 20 Hz. For the work done here, the EEG was sampled at 120 samples per second. To view, for example, a 10 Hz. alpha wave the temporal resolution provided by the 120 Hz. sampling rate, seemed to provide the minimum map to map continuity necessary for a visually pleasing presentation. Beta band activity could then be expected to further strain the limits of visual acceptability, implying that a higher sampling rate may even be in order. However, viewing 120 images / sec. is beyond the capabilities of human perception as mentioned above, so that real time mapping seems impractical. However, these arguments in no way preclude the practicality of viewing topographic maps in slow motion at 10 images / sec. in movie fashion.

If a real time display were to be a major goal, one alternative would be to amplitude demodulate a selected rhythmicity and present a sequence of maps at a reduced temporal resolution of 10 images / sec. owing to the lower frequency of such a signal. Examination of this approach is beyond the scope of this thesis, and left to the reader for further consideration.

Developmental work for this thesis also produced an alternative display format which has been mentioned earlier in this report. A page of 18 sequential maps are displayed on the color graphics monitor at once and viewed in the same way that a page of text is read. This page display facilitates a more detailed examination of the spatio-temporal variations in a sequence of topographic maps.

5.4 Future prospects

It has been mentioned, that the mapping techniques described in this thesis required a processing time of 21 seconds per map, using the present hardware / software resources described here. This complexity is quite unacceptable from a clinical point of view because of the long periods of time required to process the large amounts of data typically encountered; it is not reasonable to expect the clinician to wait 21 seconds for each map if a sequence of 10 to 20 maps is to be examined. One alternative which was described earlier, is to preprocess and store the maps for later viewing, which simply converts the complexity problem to one of storage, and results in a preprocessing delay of 11 hours to map 16 seconds of data. It is clear that system performance must be improved if the techniques described here are to be useful in a clinical setting. A goal of processing 10 spline interpolated source current or voltage maps in one second was put forward above as being a satisfactory performance level to be achieved. This goal, would in turn require a performance improvement of roughly 200 fold. The discussion which follows provides a general overview of the complexity problem and a prospectus for achieving the above goal via the exploitation of inherent parallelism given a VMEbus connection.

5.4.1 System complexity

The topographic mapping process described in this thesis involves a sequence of temporal filtering, calculating bi-cubic spline coefficients, interpolating the map and scaling the map. The most complex aspects of this sequence are the last 2, accounting for 90% of the total complexity, or roughly 19 out of 21 sec. per map. Interpolating and scaling a topographic map , required that 36 floating point operations, 2 comparisons and 1 floating point to integer conversion be performed for each pixel in a 100 x 150 pixel image. Using the big 'O' notation for describing algorithmic complexity [40], the complexity of the interpolation and scaling process is O(XY) where X and Y are the horizontal and vertical dimensions (in pixels) of the image to be produced. In comparison, the complexity of computing the spline coefficients has been discussed by Chan [4] to be O(nm) where n and m are the horizontal and vertical dimensions of the sample set. The complexity of the FIR filtering process is O(kC) where k is the number of EEG channels and C is the number of coefficients in the desired filter.

5.4.2 System performance

In the interpolation process, about 39 floating point operations (FLOPS) are performed for each of 15000 pixels, in a total of 21 seconds. Taking into account the 90% factor previously mentioned , this translates to a processing rate of about 31000 FLOPS / sec. A map generation rate of 10 maps / sec. is the minimum rate at which continuous motion could be simulated for a human observer and would supply the user with a page of 18 maps in about 2 sec. The author suggests that 2 sec. would be insignificant relative to the time required for a user to analyze 18 maps. Further benefits of this rate are: 1) storage complexity is limited to the space requirements of the raw EEG since storage of preprocessed maps would not be necessary, 2) the user need only generate maps for a particular area of interest in the raw EEG data file and 3) the user has the flexibility to change filter parameters as desired when mapping areas of interest. As mentioned the 10 map / sec. processing rate implies a need to improve the map processing rate by 200 times. This in turn, implies that a suitable mapping system should be capable of about 6.2 MFLOPS / sec.

It has been mentioned, that the sampling density of the spatial EEG is likely to increase in the future. This increase will impact on mapping complexity, so to gain a feel for the amount of this increase, table 5.1 is presented. Here, the increased complexity of a 7 x 10 and a 10 x 14 sampling grid is compared with the complexity of the 5 x 7 sampling grid of this report and an attempt has been made to keep the final map dimensions at an aspect ratio of about 100 x 150. The 5 x 7 electrode array results in a map with 24 square sections of 25 pixels a side, the 7 x 10 array in 54 square sections of 17 pixels a side, and the 10 x 14 array in 117 sections of 11 pixels a side. From this it can be seen that as sampling density increases, the map size remains relatively constant, so that the most complex portion of the mapping process changes little.

	Present 5 x 7	Future 7 x 10	Increase	Future 10 x 14	
	Increase grid. 31 electrodes	grid. 66 electrodes.	factor	grid. 136 electrodes	factor
Map generation	O(100 x 150)	O(102 x 153)	1.07	O(99 x 143)	.944
Filtering	O(31 x C)	O(66 x C)	2.13	O(136 x C)	4.39
Spline Coeff.	O(5 x 7)	O(7 x 10)	2.0	O(10 x 14)	4.0

Complexity due to increased spatial sampling

Table 5.1

As mentioned above, the time required between successive maps was empirically determined to be 21 sec. Of this time, between 1 and 2 sec. was required for temporal filtering and bi-cubic spline coefficient calculation. Since increasing the spatial sampling density impacts mainly on these 2 areas, the more pessimistic estimate of 2 sec. will be used. From the information in table 5.1, it can be shown that the time estimate for the 66 electrode array is (19 x 1.07) + (2 x 2.13) = 24.59 sec. while for the 136 electrode array, the time estimate is (19 x .944) + (2 x 4.39) = 26.72 sec. These are of course conservative estimates using the higher of the filtering-spline coefficient growth factors. This also assumes no improvement in the computation algorithms used for interpolation, filtering and calculation of spline coefficients and that the present system algorithms can transfer directly to the hardware of some prospective system. While it is somewhat simplistic, the discussion which follows should provide a rough estimate of the performance required of such a system.

Comparing the above estimates with the benchmark time of 21 sec. the computation requirement for the 66 electrode array implies a 17% complexity growth while the 136 electrode array implies a 27% complexity growth. The 66 electrode array represents a 42% increase in sampling density while the 136 electrode array represents a 100% increase in sampling density. Of course, the 136 electrode array had the advantage of producing a smaller map than the 35 electrode array. If such decrease is not considered, the complexity increase for the 136 electrode array comes to 32%. Thus, in terms of the algorithms used to date for cubic spline based mapping, a prospective system should be capable of performing 8.2 MFLOPS / sec. to achieve the above stated goal.

5.4.3 Parallelism

The subject of parallel computing is very broad, and an in depth discussion of how parallelism can be fully exploited in this application can not adequately be handled here. Indeed, such an analysis could comprise a report of its own. The discussion which follows provides an overview of the parallelism inherent in this application to show that the goals stated above are within the scope of modern microprocessor technology and a connection system such as the VMEbus.

Parallel computing was broadly categorized into 4 broad areas by Flynn [11] in 1966 and these categories are commonly used today [1,12]. Hayes [26] suggests that multiprocessors generally fall into the category which Flynn refers to as "Multiple Instruction stream Multiple Data stream" (MIMD) which covers machines which can execute more than 1 program concurrently. In the prospective system to be discussed here, a shared bus, namely a VMEbus, provides a shared interconnection structure for concurrently running processors. The advantages of this interconnection structure are low cost and the ability to add new units without revising the system structure. The disadvantages of a shared link is that excessive contention can adversely affect overall processing speed and the system is sensitive to bus failure. Thus, considerable attention must be given to the division of resources and responsibilities in order to minimize the impact of a necessarily complex arbitration process. The sensitivity to bus failure can be accepted in the interests of lowering system costs.

The VMEbus is an asynchronous specification and is completely consistent with the multiprocessor requirements to be discussed. The VMEbus structure is divided into 4 sub bus [42] structures, which are 1) the data transfer bus (DTB), comprised of up to 32 address and 32 data lines, 6 address modifier lines and assorted DTB control lines 2) the arbitration bus comprised of 6 priority specification lines and 4 daisy chained lines 3) the interrupt bus capable of 7 priorized interrupts, or daisy chained interrupts and 4) the utility bus which monitors the health of the bus. Bus arbitration monitors these lines to provide graceful shutdown during bus failure. With a 32 bit data path in the DTB and the bus clock rate, data transfer rates of up to 64 MBytes / sec. are possible. Under the VMEbus specification, one bus master must perform the bus arbitration duties, but provision is made to receive and service interrupts either through 1 supervisory processor or through 2 or more processors operating on a portion of the operating system executive.

5.4.3.1 Mapping system parallelism

A topographic mapping system under a shared connection multiprocessor system can be divided into 7 basic functions: 1) the data acquisition system, comprised of amplifiers and analog / digital conversion equipment 2) the tertiary memory system for archival data storage and system back up 3) the secondary memory system to provide quick access to mass storage of the individual subject data to be examined and system programs and files 4) the primary or operational memory system which provides the fast access necessary for the rapid processing of data 5) the color graphics display system for displaying raw data and topographic maps 6) a fast floating point computation unit capable of the 8.2 MFLOPS, or better, performance goal described above and 7) the coordinating processor. The archiving task, data acquisition task or the analysis / mapping task concurrently utilize various subsets of the 7 resources above. For the purposes of mapping, item 6 above has been shown to be of critical importance and the parallelism inherent in the computational aspects of the analysis / mapping task will be discussed separately below.

With regard to the computational complexity of the mapping problem, at least one VMEbus compatible array processor is available, which specifies a benchmark performance of 15 MFLOPS / sec. and would seem to satisfy the performance requirement for producing 10 maps / sec. Based upon an article in a recent issue of IEEE spectrum [37], an array processor would seem to provide the best compromise between MFLOP performance, cost and programmability. It is not possible to establish here the applicability of such performance specifications to the problem at hand without a detailed analysis of the architecture of available array processors, the relevance of benchmark performance measures and the high level language programmability available for an array processor. However, the mathematically intense aspects of the mapping techniques already described can be examined vis-a-vis their suitability to array processors, which would provide some assurance that such a device was being utilized to its fullest capacity.

5.4.3.2 Software parallelism

During the mapping process, from start-up to map display, the following sequence of tasks must be performed: 1) raw EEG data is displayed to the user who can scroll through in a macroscopic form, to identify sections of data to be mapped, 2) a filter with the desired bandpass characteristics must be constructed, 3) a digital filter is applied to the data, 4) b-cubic spline coefficients are determined, 5) bi-cubic polynomials are evaluated for each pixel in the map, 6) the map is scaled, and 7) the map is displayed.

In the present system, scrolling data across the screen, item 1) above, can be accomplished in software by constantly shifting data in a display buffer and reading in the updated data. However, in a prospective system the display system should eliminate host overhead for data display shifting, leaving the host the responsibility of determining updated values only. Item 2) is a one time task of relatively low complexity and can easily be performed by a host / coprocessor pair quickly. Items 3) to 6) are then performed in sequence to produce successive maps.

Convolutional application of an FIR filter, item 3), can be set up as a vector scalar product, well suited to an array processor.

Under the algorithm described in chapter 2, item 4) was broken into 2 segments. First, (m+2) and (2n) one dimensional cubic splines [O(n-2) and O(m-2) respectively] are performed on the sample set, and second, from this data a matrix operation is performed to determine the 16 bi-cubic spline coefficients. The first task is not well suited to array processing but is of low complexity. The second is highly amenable to array processing.

Evaluation of the interpolating polynomials, item 5) above, in the present system was performed using a double Horner's method, which in a sequential processing context was as time efficient and more space efficient than a matrix form of solution. However, in an array processing environment, the equations of section 3.2.1.3 can be put into a matrix form of solution whereby the cross terms $(x-x_i)^k(y-y_j)^l$ are stored as a constant 2 dimensional array to be referenced to the 16 coefficients of each bi-cubic spline in the map.

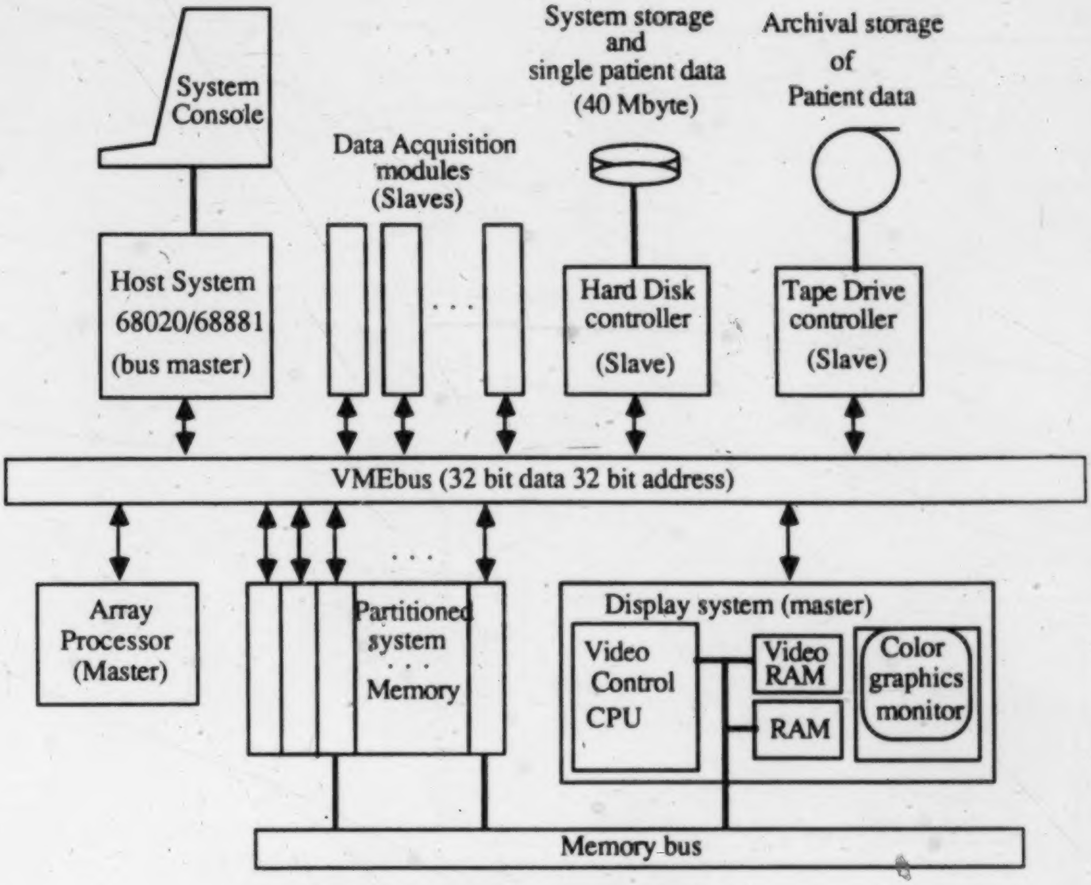
Scaling and drawing of the ellipse, item 6), is accomplished by a relation of the form

MAP(I,J) = NINT (IMAGE(I,J) - K1)K2+K3] x MASK(I,J)

where K1, K2 and K3 are scalar constants (K3 = 1.0), MASK is a 100 x 150 constant mask array to define the ellipse on the image to be displayed and IMAGE is the REAL 100 x 150 image array to be scaled. The constants K1, K2, K3 and MASK(I,J) can be stored in an array processor and the calculation set up in a pipelined manner, to take advantage of array processing architecture.

Finally, the scaled image must be placed in display memory. Ideally, the scaling operation should place the image array directly into either video memory, or memory which the display system can access independent of the host processor, eliminating unnecessary data moves.

Thus, employing an array processor would be a cost effective alternative to designing application specific hardware, as a means to remove the computational bottleneck and allow a prospective system to approach the 10 map / sec. goal. Figure 5.1 is a generalized diagram of such a prospective multiprocessor system, specific to source current mapping of the EEG and flexible enough to promote further developments. The host system is responsible for task coordination, VMEbus arbitration and basic computational requirements. The data acquisition modules could be set up as VMEbus slaves, and in the data acquisition process the host system would control data acquisition parameters on the modules, such as amplifier gain, sampling rate and storage on hard disk. As proposed, an array processor is provided for floating point intensive calculations. Main or primary system memory is shown partitioned, to facilitate multiple memory access from various VMEbus master modules within bus arbitration constraints. Having solved the computational bottle neck with the introduction of array processing, it is anticipated that contention VMEbus contention could increase due to memory references by multiple processors. Therefore, it is suggested, that as well as providing partitioning of primary memory, each memory module could provide dual access either via the VMEbus or a separate memory bus, where memory bus connections would be reconfigurable as desired. For instance, it may be desirable to set up a swinging buffer with 2 memory modules, whereby the array processor could write a map to one memory via the VMEbus, while the display processor could be displaying an image in another memory module via the memory bus.


System Console
Host System
68020/68881
(bus master)
Data Acquisition modules (Slaves)
. . .
System storage and single patient data (40 Mbyte)
Hard Disk controller
(Slave)
Archival storage of Patient data
Tape Drive controller
(Slave)
VMEbus (32 bit data 32 bit address)
Array Processor (Master)
. . .
Partitioned system
. . .
Memory
Display system (master)
Video Control CPU
Video RAM
RAM
Color graphics monitor
Memory bus

A possible multi-processor configuration
for high speed topographic mapping

Figure 5.1

5.5 Summing up

The EEG possesses a number of distinct advantages over other testing modalities such as computed X-ray tomography (CT), positron emission tomography (PET) and nuclear magnetic resonance imaging (NMR). The most significant advantage is that the EEG is the only modality which can come close to providing real time information as to brain function. In addition the cost of CT and NMR equipment is an order of magnitude higher while the cost of PET is approaching 2 orders of magnitude higher than the most sophisticated topographic mapping system modern advances can provide. Specifically, the hardware for a highly flexible development system such as the system proposed above, can probably be obtained for between $100,000 to $150,000. Such cost effectiveness is particularly significant in light of the rapidly rising costs of health care today. Further, EEG is attractive in comparison with radiographic techniques such as CT and PET in that the EEG is non-invasive (ie:- it does not expose the subject to ionizing radiation). Finally, the EEG is elicited very quickly and easily.

The developmental work described in this work represents a significant improvement in topographic mapping technology applied to date. In particular, the work presented in this report has effectively dealt with 3 major issues facing the clinical utility of EEG today. First, with the Laplacian, topographic mapping is free of the effect of the electrical recording reference. Using bi-cubic spline interpolation for interpolation , the Laplacian could be implemented analytically rather than as a numeric approximation. Second, topographic mapping of selected rhythmic components of the EEG in the source current derivation format improves the user's ability to localize generators of this activity in the cortex. With topographic mapping, the user is readily presented with the spatial aspects of these signals. In addition, bi-cubic spline interpolation provides the smoothest and sharpest topographic mapping yet presented. Finally, spatial sampling of the EEG has been discussed and while future systems will need to incorporate greater sampling densities, the system presented here improves on the sampling density used by Duffy. The characteristics of cubic spline interpolation has been discussed at some length and is established as the most effective interpolation method yet applied to topographic mapping of the EEG.

The filtering techniques used to spectrally decompose the EEG were straight forward to implement and serve to illustrate central aspects of digital filtering quite well. However, these windowing techniques, while quite effective in this application lack the sophistication of modern optimized techniques such as the Parks - McClellan algorithm [33]. It was considered outside the scope of this work to discuss these techniques in detail, as the main thrust of this thesis is topographic mapping and the issue of electrical reference. However, future systems should no doubt incorporate the most modern techniques available, to spectrally decompose the EEG.

Finally, the complexity of using bi-cubic spline interpolation has been examined vis a vis practical clinical requirements. Owing to limitations of the VME 10 computer system in place within the department of Applied Sciences in Medicine, the problem of mapping presentation utility had to be addressed by storing maps off line. This presents a large data expansion problem and is thus unacceptable from a clinical point of view. However, the VMEbus architecture has the multi-processor and data transfer capability needed to produce bi-cubic spline interpolated topographic maps at a rate of near 10 maps / sec. and thereby eliminate the need to store maps offline. This capability is particularly significant relative to the storage problem, since having it means the clinician will not necessarily need or want to map all of the EEG data available. The limitation to having the necessary capability to compute maps quickly, is one of funding rather than available technology.

REFERENCES

[1] Agrawal, Janakiram, Pathak, "Evaluating the performance of multicomputer configurations", IEEE computer, vol. 19, No. 5, pp. 23-37, May 1986.

[2] John S. Barlow, "Computerized clinical electroencephalography in perspective", IEEE Trans. Biomed. Eng., vol. BME-26, No.7, pp 377- 391, July 1979.

[3] Burden, Faires, Reynolds, Numerical Analysis. Prindle, Weber & Schmidt, Boston Massachusetts, 1978

[4] Kok Wai Chan, Image Expansion Using Interpolation and Noise cleaning methods. M.Sc. thesis, Department of Computing Science, University of Alberta, 1985

[5] J. W. Cooley, J. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. Comput., 19: pp 297-301, 1965.

[6] John E. Desmedt and Marc Bourguet, "Color imaging of parietal and frontal somatosensory potential fields evoked by stimulation of median or posterior tibial nerve in man", Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam, 62: pp 1-17, 1985.

[7] D. E. Dudgeon and R. M. Mesereau, Multidimensional digital signal processing. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.

[8] Frank H. Duffy, MD, Martha B. Denckla, MD, et. al., "Dyslexia: regional differences in brain electrical activity by topographic mapping", Annals of Neurology, Vol. 7, No. 5, May 1980.

[9] Frank H. Duffy, "Topographic display of evoked potentials: clinical applications of brain electrical activity mapping (BEAM)", Annals of the New York Acadamy of Sciences, 388: pp 183-196, 1982.

[10] Frank H. Duffy, J. L. Burchfiel, C.T. Lombroso, "Brain electrical activity mapping (BEAM): A method for extending the clinical utility of EEG and evoked potential data", Annals of Neurology, 5: pp 309-321, 1979.

[11] M.J. Flynn, "Very high-speed computing systems", Proceedings IEEE, vol. 54, pp. 1901-1909, December 1966.

[12] Geoffrey C. Fox, Paul C. Messina, "Advanced computer architectures", Scientific American, Vol. 257, No. 4, Oct. 1987, pp. 67-74.

[13] Alan S. Gevins, Robert E. Schaffer, "A critical review of electroencephalographic (EEG) correlates of higher cortical functions", CRC critical reviews in Bioeng., pp 113-164, Oct. 1980.

[14] Alan S. Gevins, Joseph C. Doyle et. al., "Electrical potentials in Human Brain during cognition: new method reveals dynamic patterns of correlation", Science magazine, Vol. 213, pp 918-922, August 1981.

[15] Alan S. Gevins, Robert E. Schaffer, et. al., "Shadows of thought: shifting lateralization of human brain electrical patterna during brief visuomotor task", Science magazine, Vol. 220, pp 97-99, April 1983.

[16] Alan S. Gevins, Nelson H. Morgan, et. al., "Human neuroelectric patterns predict performance Accuracy", Science magazine, Vol. 235, pp 580-585, Jan. 1987.

[17] Alan S. Gevins, Joseph C. Doyle, et. al., "Neurocognitive pattern analysis of a visuospatial task: rapidly-shifting foci of evoked correlations between electrodes", Psychophsiology, Vol. 22, No.1, pp 32-35, Jan. 1985.

[18] Alan S. Gevins, Joseph C. Doyle, et. al., "Lateralized cognitive processes and the electroencephalogram", Science magazine, Vol. 207, pp 1005-1008, Feb. 1980.

[19] A. S. Gevins, G. M. Zeitlin, J. C. Doyle, et. al., "Electroencephalogram correlates of higher cortical functions", Science magazine, Vol 203, pp 665-668, Feb 1979.

[20] A. S. Gevins, G. M. Zeitlin, C. D. Yingling, et. al., "EEG patterns during 'cognitive' tasks. I. Methodology and analysis of complex behaviors", Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam, 47: pp 693-703, 1979.

[21] A. S. Gevins, C. L. Yeager, G. M. Zeitlin, et. al., "On-line computer rejection of EEG artifact", Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam, 42: pp 267-274, 1977.

[22] Alan S. Gevins, "Personal communication. Progress in Topographic Mapping of Neurophysiologic Dat, F. H. Duffy (dir.)", Harvard Medical School, Oct. 16-17, 1984.

[23] Pierre Gloor, "Neuronal Generators and the problem of localization in electroencephalograph: application of volume conductor theory to electroencephalography", Journal of Clinical Neurophysiology, Raven Press, New York, Vol. 2, No. 4, pp 327-354, 1985.

[24] Pierre Gloor (trans.), HANS BERGER ON THE ELECTROENCEPHALOGRAM OF MAN: The fourteen original reports on the human electroencephalogram, Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam/London/New York, 1969, Supplement No. 28.

[25] B. Gold and K. L. Jordan Jr., "A note on digital filter synthesis", IEEE proceedings, pp 1717-1718, October 1968.

[26] John P. Hayes, Computer architecture and Organization, McGraw Hill book company, 1978

[27] S. Haykin, Editor, Topics in applied physics: Nonlinear Methods of spectral analysis, Springer-Verlag, 1983.

[28] Jack E. Hemenway, "Powerful VME bus features ease high-level µC applications", Electronics design news, pp 158-165, Jan. 1984.

[29] R. D. Katznelson, "EEG recording, electrode placement, and aspects of generator localization", In: Electric Fields of the Brain: the neurophysics of the EEG, Paul L. Nunez (ed.), Oxford University Press, 1981.

[30] D. Lehmann, "Multichannel topography of human alpha EEG fields", Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam, pp 439-449, 1971.

[31] D. Lehmann, W. Skrandies, "Reference-free identification of components of checkerboard-evoked multichannel potential fields", Electroenceph. clin. Neurophysiol., Elsevier Publishing Co., Amsterdam, pp 609-621, 1980.

[32] Lonnie C. Ludeman, Fundamentals of Digital Signal Processing, Harper & Row, Publishers, New York, 1986.

[33] J. H. McClellan, T. W. Parks, L. R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters", IEEE transactions Electroacoustics, vol AU-21, pp 506-526, Dec. 1983.

[34] Ken Nagata, Masahiro Mizukami, et. al., "Topographic electroencephalographic study of cerebral infarction using computed mapping of the EEG", Journal of Cerebral Blood Flow and Metabolism, Raven Press, New York, 2: pp 79-88, 1982.

[35] Paul L. Nunez, "The physics-EEG interface", In: Electric Fields of the Brain: the neurophysics of the EEG, Paul L. Nunez (ed.), Oxford University Press, 1981.

[36] Francois Perrin, Oliver Bertrand, and Jacques Pernier, "Scalp current density mapping: value and estimation from potential data", IEEE Trans. Biomed. Eng., vol. BME-34, No. 4, pp 283-288, April 1987.

[37] Ken Rauch, "Math chips: how they work", IEEE Spectrum, Vol. 24, No. 7, pp 5-30, July 1987.

[38] Harold W. Shipton, "EEG analysis: a history and prospectus", Annu. Rev. Biophys. Bioeng., vol. 4, pp 1-13, 1975.

[39] H. Späth, Spline algorithms for curves and surfaces, Utilitas Mathematica, Winnipeg, 1974.

[40] Donald F. Stanat, David F. McAllister, DISCRETE MATHEMATICS IN COMPUTER SCIENCE, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

[41] William D. Stanley, Gary R. Dougherty, Ray Dougherty, Digital Signal Processing, second edition, Reston Publishing Co., Inc., Reston Virginia, 1984.

[42] VMEbus specification manual, Rev. B, VMEbus manufacturers group, Aug. 1982.

APPENDIX A

A number of utility subroutines have been developed on the VME 10 microprocessor development system for interfacing with various devices attached to the system's VMEbus. The subroutines have been grouped into libraries appropriate to the device supported and for any of them to be used, the appropriate library must be linked with the main program making the call; subroutines and their library categories relevant to this project are:

GRLIB	color graphics system subroutine library

GRGET	get color graphics board
GRRLS	release color graphics board
GRBLNK	blank screen
GRCOL	graphics board color table selection
GRBAR	draw a vertical color bar
GRBARH	draw a horizontal color bar
GRDXY	display an integer*2 format image
GRDXYN	display a nibble format image
GRTEXT	write text under graphics mode
GRPXL	display a pixel array
GRLINE	vector
GRZOOM	use of graphics processor zoom

CTGLIB	contiguous file subroutine library (disk)

CTGCMD	file handling service commands for contiguous files on (allocate, assign, close, open)
CTGRDR	random record read of a contiguous file

FFTLIB	fast fourier transform library
FFTGET	get SPV 100 board (256 point FFT)
FFTRLS	release SPV 100 board
FFTC	perform complex 256 point FFT
STLIB	streaming tape subroutine library
STINIT	assign streaming tape drive
STCMD	streaming tape commands (rewind, write file mark, and other tape motion commands)
STRD	file read
STWR	file write
DIVLIB	diversified subroutine library
INTNIB	Integer*2 to nibble format image conversion
SCLI2	scale a topographic map to Integer*2 format
EVALV	evaluate bi-cubic spline scalp potential polynomials
EVALI	evaluate bi-cubic spline Laplacian polynomials

Subroutine descriptions for each of the above follow.

A.1 GRLIB

These are utility subroutines which interface with the DY-4 color graphics board. The board is based on the NEC μPD 7220 color graphics processor and provides for a pallet of 4096 colors. At any one time, 16 of these colors can be selected and placed in the look-up table. Of these 16 colors, one must always be the black background color, so that practically speaking, color selection is limited to 15 colors.

SUBROUTINE NAME: GRGET

PURPOSE: Allocates the graphics board memory

CALLING SEQUENCE: In FORTRAN;

```
        CALL GRGET
```

In ASSEMBLER;

```
        BSR GRGET
```

PARAMETERS PASSED: NONE

DESCRIPTION: Allocates the graphics board memory (physical address $C00000 and $FFDC00) to the current task. GRGET must be used before any other graphics call.

EXAMPLE:

```
    .
    CALL GRGET
    .
```

SUBROUTINE NAME: GRRLS

PURPOSE: Releases the graphics board memory

CALLING SEQUENCE: In FORTRAN;

```
        CALL GRRLS
```

In ASSEMBLER;

```
        BSR GRRLS
```

PARAMETERS PASSED: NONE

DESCRIPTION: Deallocates the graphics board memory from the current task. No other graphics call will work after GRRLS unless the memory is reallocated with GRGET.

EXAMPLE:

```
        .
        CALL GRRLS
        .
```

SUBROUTINE NAME: GRBLNK

PURPOSE: Blanks the color graphics screen

CALLING SEQUENCE: In FORTRAN;

```
CALL GRBLNK
```

In ASSEMBLER;

```
BSR GRBLNK
```

PARAMETERS PASSED: NONE

DESCRIPTION: Fills the color graphics screen with black, which must always be the color in LUT(0), the first color in the 16 color look-up table.

EXAMPLE:

```
CALL GRGET
      .
      CALL GRBLNK
      .
      CALL GRRLS
```

SUBROUTINE NAME: GRCOL

PURPOSE: Sets the 16 colors for graphics and blanks the color graphics screen.

CALLING SEQUENCE: In FORTRAN;

CALL GRCOL (LUT)

In ASSEMBLER;

BSR GRCOL

PARAMETERS PASSED: Integer*4 LUT(16)

DESCRIPTION: Each of the 16 colors desired are represented by a value determined from a concatenation of blue(b), green(g), red(r) (the 3 primary colors) as in the following 16 bit binary notation: $0000rrrrggggbbbb.

Of the 16 colors selected, black (binary 0000,0000,0000,0000) must always be used and passed to the first location in the color graphics board look-up table. Thus, black and the other 15 colors selected are placed in an Integer*4 array and passed to subroutine GRCOL, which places the 16 colors in the graphics board look-up table.

EXAMPLE: LUT(1) = 0 (black)

LUT(2) = 16 - 1 (highest intensity blue)
LUT(3) = 256 - 16 - 1 (highest intensity green)

.

LUT(15) = 4096 - 256 - 1 (highest intensity red)
LUT(16) = 4096 - 1 (highest intensity white)

.

CALL GRGET The intensity of each primary color is controlled
. by a 4 bit binary integer. Thus, the pallet of
CALL GRCOL (LUT) available colors contains 2^4 x 2^4 x 2^4 = 4096.
. Red intensity is represented by the 4 most
CALL GRRLS significant digits, followed by green, then blue.

SUBROUTINE NAME: GRBAR

PURPOSE: To display a vertically oriented color bar of the 16 colors in the graphics board look-up table (black at the bottom and color 16 at the top) at any desired screen location.

CALLING SEQUENCE: In FORTRAN;

CALL GRBAR (IX,IY,ZOOM)

In ASSEMBLER;

BSR GRBAR

PARAMETERS PASSED: Integer*4 IX,IY,ZOOM

where IX is the horizontal screen location, $0 \leq IX \leq 1280$, as measured from the left edge of the screen; IY is the vertical screen location, $0 \leq IY \leq 480$, as measured from the top edge of the screen; ZOOM is the desired screen zoom factor (1, 2, 3, 4).

DESCRIPTION: The vertical bar is enlarged as desired by the ZOOM and is displayed at location (IX,IY) on the screen, where (IX,IY) are the coordinates of the NW corner of the bar.

EXAMPLE: CALL GRGET

.

IX = 640 Places and unzoomed version of the color bar on the
IY = 240 color graphics screen, with the NW corner of the bar
ZOOM = 1 in the middle of the screen.
CALL GRBAR (IX,IY,ZOOM)

.

CALL GRRLS

SUBROUTINE NAME: GRBARH

PURPOSE: To display a horizontally oriented color bar of the 16 colors in the graphics board look-up table (black at the left and color 16 at the right) at any desired screen location.

CALLING SEQUENCE: See GRBAR

PARAMETERS PASSED: See GRBAR

DESCRIPTION: The horizontal bar is enlarged as desired by the ZOOM and is displayed at location (IX,IY) on the screen, where (IX,IY) are the coordinates of the NW corner of the bar.

EXAMPLE:

```
CALL GRGET
.
        IX = 640      Places and unzoomed version of the color bar on the
        IY = 240      color graphics screen, with the NW corner of the bar
        ZOOM = 1      in the middle of the screen.
        CALL GRBARH (IX,IY,ZOOM)
.
        CALL GRRLS
```

SUBROUTINE NAME: GRDXY

PURPOSE: Displays an image array of specified dimensions, and which is color coded to the 16 available color values, at a specified location on the color graphics screen. The image array has been formatted as Integer*2 color values for each pixel.

CALLING SEQUENCE: In FORTRAN;

```
INTEGER*4   IMAGE,IX,IY,X,Y
INTEGER*2   IMAGE1(IX,IY)
EQUIVALENCE  (IMAGE,IMAGE1)

CALL GRDXY (IMAGE,IX,IY,X,Y)
```

In ASSEMBLER;

```
BSR GRDXY
```

PARAMETERS PASSED:

```
INTEGER*4   IMAGE,IX,IY,X,Y
INTEGER*2   IMAGE1(IX,IY)
EQUIVALENCE  (IMAGE,IMAGE1)
```

where IMAGE1 is an Integer*2 array of pixel color values. The equivalence statement is necessary since the ABSOFT version of FORTRAN 77 cannot pass Integer*2 parameters. IX is the horizontal dimension of the image, $0 \le IX \le 640$; IY is the vertical dimension of the image, $0 \le IY \le 480$; X is the screen location of the left edge of the image $0 \le X \le 640$; Y is the screen location of the top edge of the image $0 \le Y \le 480$.

DESCRIPTION: The array IMAGE1 contains values representing pixel colors 0 to 15 and has the same dimensions as the desired image. The NW corner of the image is displayed at the screen location (X,Y).

```
EXAMPLE:   INTEGER*4   IMAGE,IX,IY,X,Y
           INTEGER*2   IMAGE1(64,96)
           EQUIVALENCE   (IMAGE,IMAGE1)
               .
           IX = 64      Displays a 64 x 96 pixel image in the middle of the
                        color graphics screen, with the NW corner of the
           IY = 96      image at the relative screen location (272,208).

           X = 272
           Y = 208
               .
           CALL GRGET
               .
           CALL GRDXY (IMAGE,IX,IY,X,Y)
               .
           CALL GRRLS
```

SUBROUTINE NAME: GRDXYN

PURPOSE: Displays an image array of specified dimensions, and which is color coded to the 16 available color values, at a specified location on the color graphics screen. The image array has been formatted as NIBBLE color values for each pixel in an Integer*4 array.

CALLING SEQUENCE: In FORTRAN;

```
        INTEGER*4   IMAGE(IX/8,IY),X,Y

        CALL GRDXYN (IMAGE,IX,IY,X,Y)
```

In ASSEMBLER;

```
        BSR GRDXYN
```

PARAMETERS PASSED: INTEGER*4 IMAGE(IX/8,IY),X,Y

where IMAGE is an Integer*4 array where pixel color values are represented a NIBBLES in long word integers, to conserve storage space. IX is the horizontal dimension of the image, $0 \leq IX \leq 640$; IY is the vertical dimension of the image, $0 \leq IY \leq 480$; X is the screen location of the left edge of the image $0 \leq X \leq 640$; Y is the screen location of the top edge of the image $0 \leq Y \leq 480$.

DESCRIPTION: The array IMAGE contains values representing pixel colors 0 to 15. Each BYTE in IMAGE contains 2 pixel color values. The right NIBBLE (half BYTE) of each BYTE is displayed first, followed by the left nibble. The NW corner of the image is displayed at the screen location (X,Y). For more discussion on how the NIBBLE formatted array is created, see the subroutine description for INTNIB in library DIVLIB.

```
EXAMPLE:   INTEGER*4  IMAGE(IX/8,IY),X,Y
                        .
                IX = 64        Displays a 64 x 96 pixel image that has been
                IY = 96        formatted in NIBBLES, in the middle of the color
                               graphics screen, with
                X = 272        the NW corner of the image at the relative screen
                               location (272,208).

                Y = 208
                CALL GRGET
                        .
                CALL GRDXYN (IMAGE,IX,IY,X,Y)
                        .
                CALL GRRLS
```

SUBROUTINE NAME: GRTEXT

PURPOSE: Display a string of alpha numeric characters at any location on the color graphics screen.

CALLING SEQUENCE: In FORTRAN;

```
CALL GRTEXT (X,Y,BUF,LEN,DIR,COL,ZOOM)
```

In ASSEMBLER;

```
    BSR GRTEXT
```

PARAMETERS PASSED: INTEGER*4 X,Y,BUF,LEN,DIR,COL,ZOOM

where:

(X,Y) are the coordinates of the relative screen location for the NW corner of the string, $0 \le X \le 640$ and $0 \le Y \le 480$

BUF is a buffer containing the string of ASCII characters to be displayed

LEN is the length of the character string BUF

DIR is the desired writing direction (0 for horizontal, 1 for vertical)

COL is the desired color in the look-up table (0 to 15)

ZOOM is the desired zoom factor

DESCRIPTION: This subroutine is intended to be used when the color graphics board is in graphics mode as opposed to character mode. The calling program creates an ASCII string of length LEN, which is displayed in the form specified by the above parameters.

```
EXAMPLE:    CHARACTER*80 BUF
                  INTEGER*4   X,Y,LEN,DIR,COL,ZOOM
                  .
                  CALL GRGET
                  .
                  X = 0                    String to be written horizontally on the color
                  Y = 0                    graphics screen, beginning at the NW corner
                  DIR = 0                  of the screen, using the smallest zoom factor.
                  ZOOM = 0                 The string is to be written in the 15th color of


                  LEN = 80                 the look-up table.
                  COL = 15
2                 FORMAT (A80)             The character string entered can be up to 80
                  READ(9,2) BUF            characters, entered from the system terminal.
                  CALL GRTEXT (X,Y,BUF,LEN,DIR,COL,ZOOM)
                  .
                  CALL GRRLS
```

SUBROUTINE NAME: GRPXL

PURPOSE: To display an number of pixel in the same color, on the color graphics monitor.

CALLING SEQUENCE: In FORTRAN;

```
        CALL GRPXL (ARRAY,COLOR,PIXELS)
```

In ASSEMBLER;

```
        BSR GRPXL
```

PARAMETERS PASSED: INTEGER*4 ARRAY(LEN),COLOR,PIXELS

ARRAY is a one dimensional array of relative screen addresses. The color graphics screen has a horizontal dimension of 640 pixels and a vertical dimension of 480 pixels.

LEN is the length of ARRAY

COLOR is the desired display color (0 to 15)

PIXELS is the number of pixels to be displayed and is equal to LEN

DESCRIPTION: Each pixel address is a single 2 BYTE integer relative to the NW corner of the color graphics monitor (address 0). Thus the address of the last pixel in the first row is 1279, that of the first pixel in the 2nd row is 1280 and that of the SE corner pixel is 1280 x 480 -1 = 614399.

EXAMPLE: INTEGER*4 ARRAY(2),COLOR,PIXELS

```
        .
        CALL GRGET
        .
        ARRAY(1) = 512      A pixel near the middle of the first line, and
        ARRAY(2) = 1792     the pixel immediately below it in the the
        COLOR = 11          second line are illuminated with color 11 in
        PIXELS = 2          the look-up table.
        CALL GRPXL (ARRAY,COLOR,PIXELS)
        .
        CALL GRRLS
```

SUBROUTINE NAME: GRLINE

PURPOSE: To draw a straight line between any two specified points on the color graphics monitor.

CALLING SEQUENCE: In FORTRAN;

```
        CALL GRLINE (X1,Y1,X2,Y2,COL)
```

In ASSEMBLER;

```
        BSR GRLINE
```

PARAMETERS PASSED: INTEGER*4 X1,Y1,X2,Y2,COL

(X1,Y1) are the screen coordinates of the starting point relative to the NW corner of the screen; (0,0) ≤ (X1,Y1) ≤ (640,480).

(X2,Y2) are the screen coordinates of the end point relative to the NW corner of the screen; (0,0) ≤ (X2,Y2) ≤ (640,480).

COL is the desired color from the look-up table.

EXAMPLE:

```
INTEGER*4    X1,Y1,X2,Y2,COL
            .
        CALL GRGET
            .
        COL = 8
        X1 = 0
        Y1 = 0
        X2 = 319
        Y2 = 239
        CALL GRLINE (X1,X2,Y1,Y2,COL)
            .
        CALL GRRLS
```

SUBROUTINE NAME: GRZOOM

PURPOSE: Changes the screen zoom factor.

CALLING SEQUENCE: In FORTRAN;

```
CALL GRZOOM (ZOOM)
```

In ASSEMBLER;

```
BSR GRZOOM
```

PARAMETERS PASSED: INTEGER*4 ZOOM

ZOOM is the desired zoom factor (1 to 8)

DESCRIPTION: Each pixel in an image displayed before GRZOOM is called, is replicated by ZOOM times, to enlarge the image.

EXAMPLE:

```
INTEGER*4   ZOOM
            .
            CALL GRGET
            .
            ZOOM = 4                A 4 by 4 image displayed before the
            CALL GRZOOM (ZOOM)      call to GRZOOM, will be enlarged to
            .                       a 16 by 16 image after the call.
            CALL GRRLS
```

A.2 CTGLIB

These are subroutines which allow a programmer to communicate with either the 15MByte hard disk or the 620 KByte floppy disk through the file handling service(FHS), and are designed to deal with contiguous files. Disk files are considered by the FHS to have logical block sizes of 256 bytes each, regardless of which drive is to be used.

The Versados file name format is VOL:USER.CAT.FILENAME.EXT. VOL refers to the volume ID on which the file resides: VDOS specifies a file on the hard disk drive, while floppy disks, can have any 4 letter volume ID specified by the user when the disk is formatted. USER specifies a 4 digit system user ID. User 0000 is the main system ID, and all other user numbers can range up to 9999.

SUBROUTINE NAME: CTGCMD

PURPOSE: Executes FHS commands on a contiguous file

CALLING SEQUENCE: CALL CTGCMD (VOL,CMD,LUN,FLNM,ISTAT)

PARAMETERS PASSED: INTEGER*4 VOL,CMD,LUN,FLNM,ISTAT

VOL specifies the volume ID (VDOS for hard disk) on which the desired file resides.

CMD specifies a FHS command code

- 128 allocates a specified number of sectors to the file.
- 64 assigns the volume to a system logical unit number (LUN)
- 4 closes a file

LUN specifies a logical unit number

FLNM specifies the desired filename

ISTAT is a FHS status code (ie: 0 for correct execution)

DESCRIPTION: The FHS command specified for the file indicated by VOL, FLNM, LUN, default user and null catalog. The default extension for the file is .DT (data file). CTGCMD returns the status of the command execution.

EXAMPLE:

```
INTEGER*4   CMD,LUN,ISTAT
            CHARACTER   FLNM*8,VOL*4

            CMD = 128             Allocates 1500 sectors to the contiguous file
            LUN = 2               SAVE:def..EEG.DT assigned to logical unit
            ISTAT = 1500                number 2.
            VOL = 'SAVE'
            FLNM = 'EEG'
            CALL CTGCMD (VOL,CMD,LUN,FLNM,ISTAT)
            IF (ISTAT.NE.0) CALL ERROR(ISTAT)
```

SUBROUTINE NAME: CTGRDR

PURPOSE: Perform a random record read on a contiguous file.

CALLING SEQUENCE: In FORTRAN;

```
CALL CTGRDR (ARRAY,LUN,BLKNO,BLOCKS,ISTAT)
```

In ASSEMBLER;

```
BSR CTGRDR
```

PARAMETERS PASSED: INTEGER*4 ARRAY,LUN,BLKNO,BLOCKS,ISTAT

ARRAY is the array into which the data file is to be read. It must be dimensioned appropriate to the data structure on the tape and sized to accommodate the number of records specified by BLOCKS.

LUN is the logical unit number assigned to the file.

BLKNO is the record number at which to commence reading the data. Counting begins at zero.

BLOCKS is the number of records to be read.

ISTAT is a value returned by CTGRDR to indicate read status after the read operation has been performed (0 = correct return; -1 = end of file; a number greater than 0 = error).

DESCRIPTION: The subroutine reads BLOCKS contiguous sectors (256 bytes / sector) from a contiguous file assigned to LUN, beginning with sector BLKNO. The data is read into ARRAY and CTGRDR returns a numeric value indicating the status of the read operation after it has been performed.

EXAMPLE:

```
INTEGER*4    ARRAY(8,64),LUN,ISTAT,BLKNO,BLOCKS
        LUN = 2             Reads 8 logical records (2048 bytes) from the
        BLKNO = 64          contiguous data file attached to logical unit
        BLOCKS = 8          number 2, beginning at record number 64.
        CALL CTGRDR (ARRAY,LUN,BLKNO,BLOCKS,ISTAT)
        IF (ISTAT.EQ.1) CALL EOF
        IF (ISTAT.NE.0) CALL ERROR (ISTAT)
```

A.3 FFTLIB

This library of subroutines is associated with the BURR-BROWN SPV 100 signal processing board attached to the VMEbus. At present, the board is only equipped with firmware to perform a 256 point complex FFT with its on board TMS 320 signal processing chip. The firmware is capable of performing the following types of FFT's:

Real signal input --- Magnitude response output
Real signal input --- Complex response output
Complex signal input --- Complex response output
Complex signal input --- Magnitude response output.

SUBROUTINE NAME: FFTGET

PURPOSE: Attaches the SPV 100 board memory to the system.

CALLING SEQUENCE: In FORTRAN;

```
        CALL FFTGET (STATUS)
```

In ASSEMBLER;

```
        BSR FFTGET
```

PARAMETERS PASSED: INTEGER*4 STATUS

STATUS is an Integer*4 variable returned by FFTGET to indicate the status of the return (0 for a correct return, 1 for an error).

DESCRIPTION: Allocates the SPV 100 board memory (beginning at address $FE0000) to the current task.

EXAMPLE:

```
INTEGER*4    STATUS
         .
         CALL FFTGET (STATUS)
         IF (STATUS.EQ.1) CALL ERROR
```

SUBROUTINE NAME: FFTRLS

PURPOSE: Releases the SPV 100 board memory.

CALLING SEQUENCE: In FORTRAN;

```
CALL FFTRLS (STATUS)
```

In ASSEMBLER;

```
BSR FFTRLS
```

PARAMETERS PASSED: See FFTGET

DESCRIPTION: Deallocates the SPV 100 board memory allocated with FFTGET. The status of the return is provided.

EXAMPLE: See FFTGET

SUBROUTINE NAME: FFTC

PURPOSE: To perform a 256 point 1-D FFT.

CALLING SEQUENCE: In FORTRAN;

```
        CALL FFTC (INPUT,OUTPUT,LEN,CTRL)
In ASSEMBLER;
        BSR FFTC
```

PARAMETERS PASSED:INTEGER*2 INPUT1(2*LEN),OUTPUT1(2*LEN)

```
INTEGER*4    INPUT,OUTPUT,LEN,CTRL
EQUIVALENCE (INPUT,INPUT1)
EQUIVALENCE (OUTPUT,OUTPUT1)
```

DESCRIPTION: Performs a 256 point FFT on the array INPUT1 in the desired format. The possible formats are those described in the FFTLIB introduction above and is specified with CTRL. The result of the FFT is placed in OUTPUT1. FFTC utilizes polling, rather than a vectored interrupt handling routine to determine when the FFT is complete.

EXAMPLE: INTEGER*4 LEN,CTRL,INPUT,OUTPUT

```
INTEGER*2 INPUT1(512),OUTPUT1(512)
EQUIVALENCE (INPUT,INPUT1)
EQUIVALENCE (OUTPUT,OUTPUT1)
      .
LEN = 256     Performs a Real input --- Complex response output
CTRL = 2      256 point FFT on INPUT1 and places the result in
      .       OUTPUT1.
CALL FFTC (INPUT,OUTPUT,LEN,CTRL)
      .
```

A.4 STLIB

This library of subroutines allow user program access to the streaming tape drive, that isn't provided through FORTRAN. The streaming tape drive is attached to the VME 10 development system with a Motorola MVME 350 VMEbus compatible module which supports the industry standard QIC-02 intelligent tape drive interface.

SUBROUTINE NAME: STINIT

PURPOSE: Assigns the streaming tape drive, (known to the operating system as device #ST40) to a logical unit number.

CALLING SEQUENCE: CALL STINIT (LUN,ISTAT)

PARAMETERS PASSED: INTEGER*4 LUN,ISTAT

LUN is the logical unit number to which the tape drive is assigned.

ISTAT is a status variable returned by the subroutine (see FHS for error values; zero indicates a normal return)

DESCRIPTION: Prepares the streaming tape drive for input / output by assigning a LUN to it. The status of the initialization operation is returned. To use this subroutine, the tape must first have been mounted under Versados using the tape mount command TMT #ST40.

EXAMPLE:
```
INTEGER*4    LUN,ISTAT

             .
             LUN = 1
             CALL STINIT (LUN,ISTAT)
             IF (ISTAT.NE.0) CALL ERROR (ISTAT)
             .
```

SUBROUTINE NAME: STCMD

PURPOSE: Executes tape commands useful to the programmer and provided by the MVME 350 board such as tape rewind, retensioning, searching for file marks, writing file marks, erasing the tape, seeking the end of the tape etc.

CALLING SEQUENCE: CALL STCMD (CODE,CMD,LUN,ISTAT)

PARAMETERS PASSED: INTEGER*4 CODE,CMD,LUN,ISTAT

ISTAT is the standard status return value
CODE is the appropriate IOS command code
CMD is the desired IOS command
LUN is the logical unit number defined with STINIT

DESCRIPTION: Executes an input / output service (IOS) command on the streaming tape drive, and returns the status of the operation. STINIT must have been used in the program before a call to STCMD.

EXAMPLE: INTEGER*4 CODE,CMD,LUN,ISTAT

```
CODE = 1     CODE and CMD define a tape rewind operation to be
CMD = 2      performed by the call to STCMD. The tape is
LUN = 1      assigned logical unit number 1 by STINIT
CALL STINIT (LUN,ISTAT)
CALL STCMD (CODE,CMD,LUN,ISTAT)
```

SUBROUTINE NAME: STRD

PURPOSE: To read a specified number of blocks of data from the streaming tape drive.

CALLING SEQUENCE: CALL STRD (ARRAY,BLOCKS,LUN,ISTAT)

PARAMETERS PASSED: INTEGER*4 ARRAY,BLOCKS,LUN,ISTAT

ARRAY is the array into which data is to be read. It must be large enough to accommodate the amount of data specified by BLOCKS. ARRAY may actually be of any variable type desired, appropriate to the data type on the tape.

LUN and ISTAT are as specified in STINIT and STCMD.

DESCRIPTION: Reads Blocks physical blocks (512 bytes / physical block) of data from the streaming tape drive into ARRAY. Before reading a block, look at ISTAT to make sure the end of the previous file has been reached.

```
EXAMPLE:   INTEGER*4    BLOCKS,LUN,ISTAT
                 CHARACTER    ARRAY*512

                 .

                 IF(ISTAT.NE.194) CALL EOF
                 BLOCKS = 1  Check to see if the EOF has been reached with the
                 LUN = 1     last tape routine used, if not find it, and read one
                 .           block of character data from the tape
                 CALL STRD (ARRAY,BLOCKS,LUN,ISTAT)
                 .
```

SUBROUTINE NAME: STWR

PURPOSE: To write a specified number of physical data blocks to the streaming tape.

CALLING SEQUENCE: CALL STWR (ARRAY,BLOCKS,LUN,ISTAT)

PARAMETERS PASSED: See STRD

DESCRIPTION: The write operation analogous to the read operation.

EXAMPLE:
```
INTEGER*4    ARRAY(256),BLOCKS,LUN,ISTAT

             BLOCKS = 2  Writes the integer data in ARRAY to the tape, which
             LUN = 1     has been assigned to logical unit number 1:
             CALL STWR (ARRAY,BLOCKS,LUN,ISTAT)
```

A.5 DIVLIB

This is a library of subroutines which are particularly relevant to the topographic mapping work in this thesis. Some, such as EVALV, EVALI and SCLI2 are subroutines previously developed and written in FORTRAN, but which have been optimized in ASSEMBLER to speed program operation.

SUBROUTINE NAME: INTNIB

PURPOSE: To convert color image arrays from Integer *2 format to nibble format for conserving space on mass storage media.

CALLING SEQUENCE: CALL INTNIB (AINT,ANIB,LEN)

PARAMETERS PASSED: INTEGER*4 AINT(N),ANIB(N/4),LEN

AINT is an array which contains the integer format image
ANIB is the output array which contains the nibble format image
LEN is a variable corresponding to the number of long words to be converted into bytes.

DESCRIPTION: The color and intensity of a pixel is coded as one of the 16 colors available in the look-up table of the color graphics board. This information can be coded in to 4 bits of information (a NIBBLE) which is 1/4 the size of the Integer*2 information typically output by the graphics routines described earlier.

For every long word (element) in AINT: the right nibble of the first word goes into the right nibble of the output byte and the right nibble of the second word goes into the left nibble of the output byte. Thus, in hex, $000X000Y becomes $YX.

EXAMPLE:

```
INTEGER*2    VECTOR(256)
             INTEGER*4    AINT,VEC(32),LEN
             EQUIVALENCE  (AINT,VECTOR)
                  .
             LEN = 128
             CALL INTNIB (AINT,VEC)
                  .
```

SUBROUTINE NAME: SCLI2

PURPOSE: To take an Integer*2 image produced by either EVALV or EVALI and scale it to the 16 color values available in the look-up table of the color graphics board. The subroutine also incorporates the ability to draw an ellipse inside the 100 by 150 pixel image, to create a rounded map reminiscent of the shape of a head.

CALLING SEQUENCE:

CALL SCLI2(BIMAGE,IMAGE,MASK,IX,IY,BMX,BMN)

PARAMETERS PASSED:

INTEGER*4 BIMAGE,IMAGE,MASK,IX,IY,BMX,BMN

BIMAGE is the Integer*4 array containing the I*2 image to be scaled

IMAGE is the output Integer*4 array for the scaled I*2 format color image

MASK is an elliptically bounded array containing 1's inside the ellipse and 0's outside the ellipse, for rounding the shape of the final displayed image.

IX an IY are the image dimensions.

BMX and BMN are the maximum and minimum values the image is to be scaled to.

DESCRIPTION: Subroutines EVALI and EVALI interpolate the electrode grid and produce a 100 by 150 array of I*2 interpolated values. SCLI2 scales these values to one of 15 levels corresponding to the 15 colors available. The scaling limits are established by BMX and BMN and image values are clipped within these limits. The color scale used in WRTAPE and RDTAPE is set up to be symmetric about color 8 in the look-up table, and BMX and BMN have the same magnitude. Finally, SCLI2 draws an ellipse around the image.

SUBROUTINE NAME: EVALV AND EVALI

PURPOSE: To evaluate the potential surface interpolating polynomials or the Laplacian of these polynomials respectively.

CALLING SEQUENCE: CALL EVALV (XX,YY,A,IMAGE)
or CALL EVALI (XX,YY,A,IMAGE)

PARAMETERS PASSED: INTEGER*2 IMAGE(100,150)
INTEGER*4 XX(5),YY(7)
REAL A(4,7,4,4)

* IMAGE is the interpolated output image.

XX and YY are the x,y coordinates of the grid electrodes.

A is the array of bi-cubic spline coefficients for the 16 rectangular patches enclosed by the 5 by 7 electrode grid.

DESCRIPTION: EVALV evaluates the bi-cubic spline polynomials specified by the coefficients in A to interpolate the potential surface, while EVALI evaluates the Laplacian of these polynomials to interpolate the source / sink currents corresponding to the potential surface. The subroutines follow a double Horner's scheme of 2-D polynomial evaluation. The subroutines have been optimized to minimize the time needed for address calculations, when indexing through the coefficient matrix A.

APPENDIX B

The following FORTRAN subroutines, adapted from Spath [39], were used in WRTAPE with some modification, to implement the cubic and bi-cubic spline interpolation algorithm described in chapter 2. The modifications to the subroutines enabled WRTAPE to use the 'free' boundary condition as well as the 'clamped' boundary condition, for interpolation.

B.1 SUBROUTINE CUBTWO

PURPOSE: To calculate the coefficients of the bi-cubic spline polynomials which interpolate a 2-D spatially sampled function under either the 'free' boundary condition ($\partial^2 S/\partial x^2 = \partial^2 S/\partial y^2 = K$) or the 'clamped' boundary condition ($\partial S/\partial x = \partial S/\partial y = K1$). CUBTWO implements the solution of the bi-cubic spline coefficients described in chapter 2.

CALLING SEQUENCE: CALL CUBTWO (N,M,X,Y,U,P,Q,R,Y2,A,BOUND)

GLOBAL PARAMETERS : INTEGER*4 N,M
REAL X(N),Y(M),Y2(⌈NM⌉),A(N-1,M-1,4,4)
REAL U(N,M),P(N,M),Q(N,M),R(N,M)
CHARACTER BOUND*1

N is the number of sample points in the x dimension

M is the number of sample points in the y dimension

X and Y give the (x,y) coordinates of the sampling grid.

U gives the input values of the sampled function to be interpolated.

P is an output array for the 'x' first partial derivatives of the interpolating spline polynomials at the grid locations. It is used to input the 'clamping' boundary conditions, but is otherwise initialized to 0's.

Q is an output array analogous to P, but is used for the 'y' first partial derivative of the interpolating spline polynomials.

R is an output array for the cross derivatives $\partial^2 S/\partial x \partial y$ of the interpolating spline polynomials at the grid locations. It is also used to input the 4 corner boundary conditions, necessary for the solution of the bi-cubic spline problem.

Y2 is an array used to input the 'free' boundary condition to the 1-D cubic spline.

LOCAL PARAMETERS: REAL ZX(5),ZY(7),F(⌈NM⌉)
REAL DX(N),DY(M)
REAL B(4,4),C(4,4),D(4,4),E(4,4)
INTEGER*4 N1,M1
CHARACTER CROSS*1

DX and DY are the values $1/\Delta x_i$ and $1/\Delta y_j$ respectively as in equations 2.9.

B,C,D and E are used in the matrix solution of A from the matrices U,P,Q AND R as described in chapter 2.

N1 = N-1 and M1 = M-1.

CROSS is a character variable used to always implement the 1-D clamped boundary condition on the fourth call to PERM.

SUBROUTINES CALLED: CUBA,PERM,MATRIX

FORTRAN LISTING:

```
      SUBROUTINE CUBTWO (N,M,X,Y,U,P,Q,R,Y2,A,BOUND)
C
C Variable declarations
C
      INTEGER*4  N,M,N1,M1
      REAL  X(N),Y(M),Y2(⌈NM⌉),A(N1,M1,4,4)
      REAL  U(N,M),P(N,M),Q(N,M),R(N,M),Y2(⌈NM⌉)
      REAL  ZX(N),ZY(N),F(⌈NM⌉),ORD(⌈NM⌉),UNB(⌈NM⌉)
      REAL  DX(N),DY(M)
      REAL  B(4,4),C(4,4),D(4,4),E(4,4)
      CHARACTER  BOUND*1,CROSS*1
C
C Initialization
C
      CROSS = 'C'
      N1 = N-1
      M1 = M-1
      DO 1 I = 1,N1
         DX(I) = 1./(X(I+1) - X(I))
1     CONTINUE
      DO 2 J = 1,M1
         DY(J) = 1./(Y(J+1) - Y(J))
2     CONTINUE
```

```
      DO 3 I = 1,4
      DO 3 J = 1,4
         B(I,J) = 0.
         E(I,J) = 0.
3     CONTINUE
      B(1,1) = 1.
      B(2,2) = 1.
      E(1,1) = 1.
      E(2,2) = 1.
C
C Calculate the bi-cubic spline derivatives
C
      IF (N.EQ.2.AND.M.EQ.2) GO TO 8
      IF (N.EQ.2) GO TO 5
      IF (BOUND.EQ.'C') THEN
         CALL CUBA (N1,DX,ZX)
      ENDIF
      IVJ = 0
      CALL PERM (IVJ,M,1,N,N1,P,U,ORD,UNB,DX,ZX,BOUND,Y2)
5     IF (M.EQ.2) GO TO 6
      IF (BOUND.EQ.'C') THEN
         CALL CUBA (M1,DY,ZY)
      ENDIF
      IVJ = 1
      CALL PERM (IVJ,N,1,M,M1,Q,U,ORD,UNB,DY,ZY,BOUND,Y2)
6     IF (N.EQ.2) GO TO 7
      IVJ = 0
      CALLPERM(IVJ,M,M1,N,N1,R,Q,ORD,UNB,DX,ZX,BOUND,Y2)
7     IF (M.EQ.2) GO TO 8
      IF (BOUND.EQ.'F') THEN
         CALL CUBA (M1,DY,ZY)
      ENDIF
      IVJ = 1
      CALL PERM (IVJ,N,1,M,M1,R,P,ORD,UNB,DY,ZY,CROSS,Y2)
C
C Now that U,P,Q AND R have been determined, calculate the bi-cubic spline
C coefficients.
C
```

```
8     DO 16 I = 1,N1
         I1 = I+1
         CALL MATRIX (DX(I),B)
         DO 15 J = 1,M1
            J1 = J+1
            C(1,1) = U(I,J)
            C(1,2) = Q(I,J)
            C(2,1) = P(I,J)
            C(2,2) = R(I,J)
            C(1,3) = U(I,J1)
            C(1,4) = Q(I,J1)
            C(2,3) = P(I,J1)
            C(2,4) = R(I,J1)
            C(3,1) = U(I1,J)
            C(3,2) = Q(I1,J)
            C(4,1) = P(I1,J)
            C(4,2) = R(I1,J)
            DO 10 K1 = 1,4
            DO 10 K2 = 1,4
               SUM = 0.
               DO 9 K = 1,4
                  SUM = SUM+B(K1,K)*C(K,K2)
9              CONTINUE
               D(K1,K2) = SUM
10          CONTINUE
            CALL MATRIX (DY(J),E)
            DO 13 K1 = 1,4
            DO 13 K2 = 1,4
               SUM = 0.
               DO 12 K = 1,4
                  SUM = SUM+D(K1,K)*E(K2,K)
12             CONTINUE
               A(I,J,K1,K2) = SUM
13          CONTINUE
15       CONTINUE
16    CONTINUE
      RETURN
      END
```

B.2 SUBROUTINE CUBA

PURPOSE: To perform diagonalization of the tri-diagonal coefficient matrix for the system of equations 2.9, under the 'clamped' boundary condition. Since the coefficient matrix does not change for application of the 1-D cubic spline to successive rows or columns of the sample grid, diagonalization need only be performed once. Subroutines CUBA and CUBB together perform the same function as CUBICI.

CALLING SEQUENCE: CALL CUBA (N1,DX,ZX)

GLOBAL PARAMETERS: INTEGER*4 N1
REAL DX(N),ZX(N)

N is the number of 1-D sample points to be interpolated. N1=N - 1.
DX --- See CUBTWO. ---
ZX is an output array produced by the diagonalization process and used by CUBB to complete the solution of the 1-D spline.

LOCAL PARAMETER: REAL H1,H2

H1 and H2 are working variables used for matrix diagonalization.

SUBROUTINES CALLED: NONE

FORTRAN LISTING:

```
      SUBROUTINE CUBA (N1.DX.Z)
C
C Variable declarations
C
      REAL   DX(N),Z(N)
      INTEGER*4   N1
```

```
C
C Matrix diagonalizations
C
      Z(1) = 0.
      J1 = 1
      H1 = DX(1)
      DO 1  K = 2,N1
         H2 = DX(K)
         Z(K) = 1./(2.*(H1+H2) - H1*H1*Z(J1))
         J1 = K
         H1 = H2
1     CONTINUE
      RETURN
      END
```

B.3 SUBROUTINE CUBB

PURPOSE: To complete the gauss elimination process on the simultaneous equations 2.9 once the diagonalization of CUBA has been performed. See subroutine CUBA.

CALLING SEQUENCE: CALL CUBB (N1,DX,Y,Y1,Z)

GLOBAL PARAMETERS: INTEGER*4 N1
REAL DX(N),Y(N),Y1(N),Z(N)

N1 and DX --- See CUBA.

Y is the array of 1-D ordinates (data values) to be interpolated.

Y1 is the array of cubic spline first derivatives for each data point.

Z is the working array determined in CUBA.

LOCAL PARAMETERS: REAL F(⌈NM⌉),R1,R2,H1,H2

F is a working array used in the gauss elimination of the system of equations 2.9.

R1,R2,H1 and H2 --- See CUBA

SUBROUTINES CALLED: NONE

FORTRAN LISTING:

```
      SUBROUTINE CUBB (N1,DX,Y,Y1,Z)
C
C  Variable declarations
C
      REAL    DX(N),Y(N),Y1(N),Z(N),F(N)
      INTEGER *4   N1
```

```
C
C Gauss elimination
C
      F(1) = 0.
      DO 2  K = 1,N1
         H2 = DX(K)
         R2 = 3.*H2*H2*(Y(K+1) - Y(K))
         IF (K.EQ.1) GO TO 1
         H = R1+R2
         IF (K.EQ.2)  H = H - H1*Y1(1)
         IF (K.EQ.N1)  H = H - H2*Y1(N1+1)
         F(K) = Z(K)*(H - H1*F(J1))
1        J1 = K
         H1 = H2
         R1 = R2
2     CONTINUE
      Y1(N1) = F(N1)
      IF (N1.LE.2)  RETURN
      N2 = N1 - 1
      DO  3  J1 = 2,N2
         K = N1 - J1 +1
         Y1(K) = F(K) - Z(K)*DX(K)*Y1(K+1)
3     CONTINUE
      RETURN
      END
```

B.4 SUBROUTINE PERM

PURPOSE: To perform a series of 1-D cubic spline interpolations on successive rows or columns of a 2-D sampled data set for the purposes of determining a matrix of x or y first partial derivatives.

CALLING SEQUENCE: CALL PERM (IVJ,M,M1,N,N1,P,U,DX,ZX,BOUND,Y2)

GLOBAL PARAMETERS: INTEGER*4 IVJ,N,M,N1,M1
REAL P(N,M),U(N,M)
READ DX(⌈NM⌉),ZX(⌈NM⌉),Y2(⌈NM⌉)
CHARACTER BOUND*1

N,M,N1,M1 --- as before.

IVJ is a control variable which informs PERM whether CUBTWO is calling for operations on rows or columns so that the derivative arrays P,Q and R are filled properly.

P and U --- See CUBTWO.

DX,ZX,Y2 --- See CUBTWO.

BOUND --- See CUBTWO.

LOCAL PARAMETERS: REAL ORD(⌈NM⌉),UNB(⌈NM⌉)

ORD and UNB are working arrays used by PERM to take successive rows or columns from a 2-D sampled data set, fit 1-D cubic splines to them, and return the computed 1-D spline derivatives.

SUBROUTINES CALLED: CUBB,CUBI2

FORTRAN LISTING:

```
      SUBROUTINE PERM (IVJ,M,M1,N,N1,P,U,DX,ZX,BOUND,Y2)
C
C Variable declarations
C
      INTEGER*4  IVJ,N,M,N1,M1
      REAL  P(N,M),U(N,M),DX(⌈NM⌉),ZX(⌈NM⌉),Y2(⌈NM⌉)
      CHARACTER  BOUND*1
```

```
C
C Fit 1-D cubic splines to successive rows or columns of the input data in U
C
      DO 4  J = 1,M,M1
         DO 2  I = 1,N
            IF (IVJ.NE.0) GO TO 1
            ORD(I) = U(I,J)
            UNB(I) = P(I,J)
            GO TO 2
1           ORD(I) = U(J,I)
            UNB(I) = P(J,I)
2        CONTINUE
         IF (BOUND.EQ.'C') THEN
            CALL CUBB (N1,DX,ORD,UNB,ZX)
         ELSEIF (BOUND.EQ.'F') THEN
            CALL CUBI2 (N,DX,ORD,UNB,Y2)
         ENDIF
         IF (BOUND.EQ.'C') THEN
            IST = 2
            IFF = N1
         ELSEIF (BOUND.EQ.'F') THEN
            IST = 1
            IFF = N
         ENDIF
         DO  3  I = IST,IFF
            IF (IVJ.EQ.0)  P(I,J) = UNB(I)
            IF (IVJ.NE.0)  P(J,I) = UNB(I)
3        CONTINUE
4     CONTINUE
      RETURN
      END
```

B.5 SUBROUTINE MATRIX

PURPOSE: Assist CUBTWO in the matrix solution of the coefficients A(I,J,K,L).

CALLING SEQUENCE: CALL MATRIX (DX(I),B)

GLOBAL PARAMETERS: REAL B(4,4),DX(I)

B is a working matrix mentioned in CUBTWO, which is involved in the matrix solution of the bi-cubic spline coefficient matrix A(I,J,K,L).

DX(I) --- See CUBTWO.

LOCAL PARAMETERS: REAL SUM

SUM is a working variable to simplify the expression DX(I)*DX(I)

SUBROUTINES CALLED: NONE

FORTRAN LISTING:

```
      SUBROUTINE MATRIX (H,B)
C
C Variable declarations
C
      REAL    H,B(4,4),SUM
C
C Matrix formation
C
      SUM = H*H
      B(3,1) = - 3.*SUM
      B(3,2) = - 2.*H
      B(3,3) = - B(3,1)
      B(3,4) = - H
      B(4,1) = 2.*H*SUM
      B(4,2) = SUM
      B(4,3) = -B(4,1)
      B(4,4) = SUM
      RETURN
      END
```

B.6 SUBROUTINE CUBICI

PURPOSE: To implement the solution of equations 2.9 under the 'clamped' boundary condition. CUBA and CUBB together perform the same function as CUBICI.

CALLING SEQUENCE: CALL CUBICI (N,DX,Y,Y1)

GLOBAL PARAMETERS: INTEGER*4 N
REAL DX(N),Y(N),Y1(N)

N --- As before.

DX --- See CUBTWO.

Y are the ordinates of the sampled data set.

Y1 is the output array for the first derivatives of the cubic spline fitted to the points (x,y), at the sample points x. It can also be used to input the 'clamping' boundary conditions.

LOCAL PARAMETERS: REAL F(N),G(N),Z,H,H1,H2,R1,R2

SUBROUTINES CALLED: NONE

FORTRAN LISTING:

```
      SUBROUTINE CUBICI (N,X,Y,Y1)
C
C Variable declarations
C
      INTEGER*4  N,N1
      REAL    X(N),Y(N),Y1(N),F(N),G(N),Z,H,H1,H2,R1,R2
```

```
C
C Coefficient matrix diagonalization
C
        N1 = N - 1
        G(1) = 0.
        F(1) = 0.
        DO  2  K = 1,N1
           J2 = K+1
           H2 = 1./(X(J2)-X(K))
           R2 = 3.*H2*H2*(Y(J2) - Y(K))
           IF (K.EQ.1)  GO TO 1
           Z = 1./(2.*(H1+H2) - H1*G(J1))
           G(K) = Z*H2
           H = R1+R2
           IF (K.EQ.2)  H = H - H1*Y1(1)
           IF (K.EQ.N1)  H = H - H2*Y1(N)
           F(K) = Z*(H - H1*F(J1))
1          J1 = K
           H1 = H2
           R1 = R2
2       CONTINUE
C
C Evaluate the solution
C
        Y1(N1) = F(N1)
        IF (N1.LE.2)  RETURN
        N2 = N1 - 1
        DO  3  J1 = 2,N2
           K = N - J1
           Y1(K) = F(K) - G(K)*Y1(K+1)
3       CONTINUE
        RETURN
        END
```

B.7 SUBROUTINE CUBI2

PURPOSE: To implement the solution of equations 2.9 under the 'free' boundary condition.

CALLING SEQUENCE: CALL CUBI2(N,DX,Y,Y1,Y2)

GLOBAL PARAMETERS: INTEGER*4 N
REAL DX(N),Y(N),Y1(N),Y2(N)

N --- As before.
DX --- See CUBTWO.
Y are the ordinates of the sampled data set.
Y1 is the output array for the first derivatives of the cubic spline fitted to the points (x,y), at the sample points x.
Y2 is the input array for the 'free' boundary condition.

LOCAL PARAMETERS: REAL F(N),G(N),Z,H,H1,H2,R1,R2

SUBROUTINES CALLED: NONE

FORTRAN LISTING:

```
      SUBROUTINE CUBI2 (N,DX,Y,Y1,Y2)
C
C Variable declarations
C
      INTEGER*4    N,N1,N2
      REAL     DX(N),Y(N),Y1(N),Y2(N),F(N),G(N),Z,H,H1,H2,R1,R2
```

```
C
C  Coefficient matrix diagonalization
C
        N1 = N - 1
        J1 = 1
        H1 = 0.
        F(1) = 0.
        G(1) = 0.
        R1 = (Y2(1)/2.)*(-1.)
        DO  3  K = 1,N
           IF (K.LE.N1)  GO TO 1
           H2 = 0.
           R2 = Y2(N)/2.
           GO TO 2
1          J2 = K+1
           H2 = DX(K)
           R2 = 3.*H2*H2*(Y(J2) - (Y(K))
2          Z = 1./(2.*(H1+H2) - H1*G(J1))
           G(K) = Z*H2
           F(K) = Z*((R1+R2) - H1*F(J1))
           J1 = K
           H1 = H2
           R1 = R2
3       CONTINUE
C
C  Evaluate the solution
C
        Y1(N) = F(N)
        DO  4  J1 = 1,N1
           K = N - J1
           Y1(K) = F(K) - G(K)*Y1(K+1)
4       CONTINUE
        RETURN
        END
```

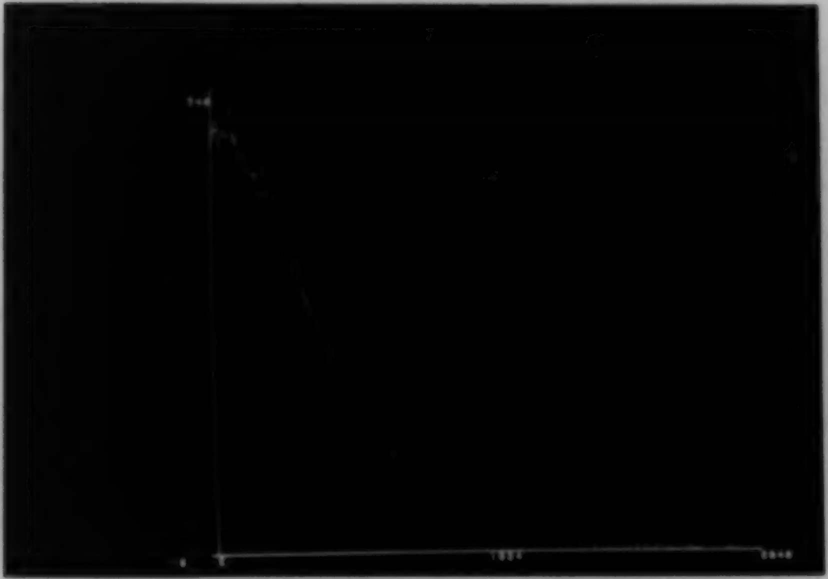
APPENDIX C

The following set of photographs serve to illustrate feedback information provided to the user by the interactive programs WRTAPE and RDTAPE.

Figure C.1 illustrates a histogram of data values for a large section of raw data. This display is particularly useful for the user when setting up to test a subject, so as to choose an amplifier gain setting which will allow optimal utilization of the A/D dynamic range.

Figure C.2 is information from the tape header file which informs the user about the information contained on the tape.

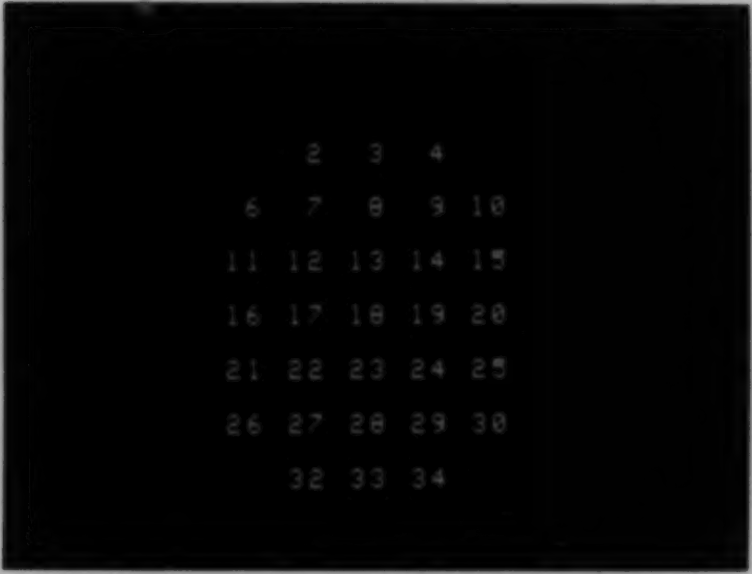
Figure C.3 illustrates the electrode configuration, so that the user may select which 8 channels of raw EEG data he/she may wish to observe, when selecting a section of maps to observe.



WRTAPE histogram

Figure C.1

RDTAPE header file information
Figure C.2



```
      2   3   4
  6   7   8   9  10
 11  12  13  14  15
 16  17  18  19  20
 21  22  23  24  25
 26  27  28  29  30
     32  33  34
```

RDTAPE channel numbering pattern
Figure C.3

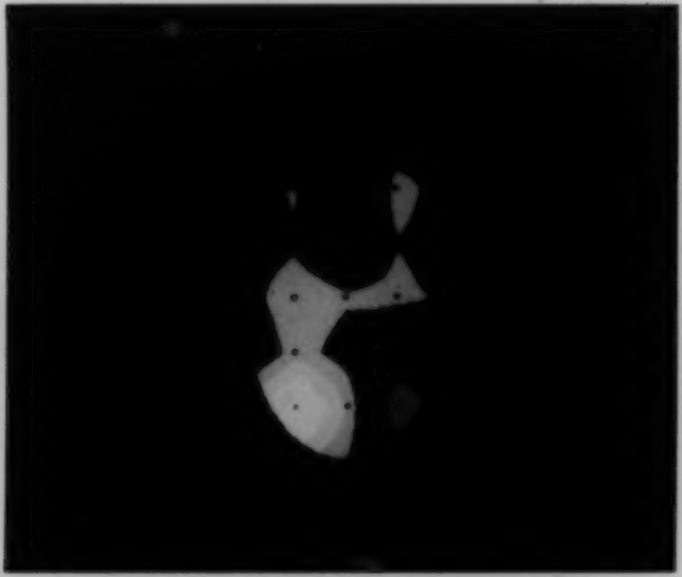
Figure C.4 is simply an illustration of 8 channels of raw EEG information in the traditional polygraphic presentation. This provides the user with a more familiar presentation format with which to scan the data, so as to select an appropriate section of data for which he/she can view topographic maps.

Figure C.5 is an illustration of the enlarged display which is used if the user wishes to display topographic maps in a 'cartooning' fashion.

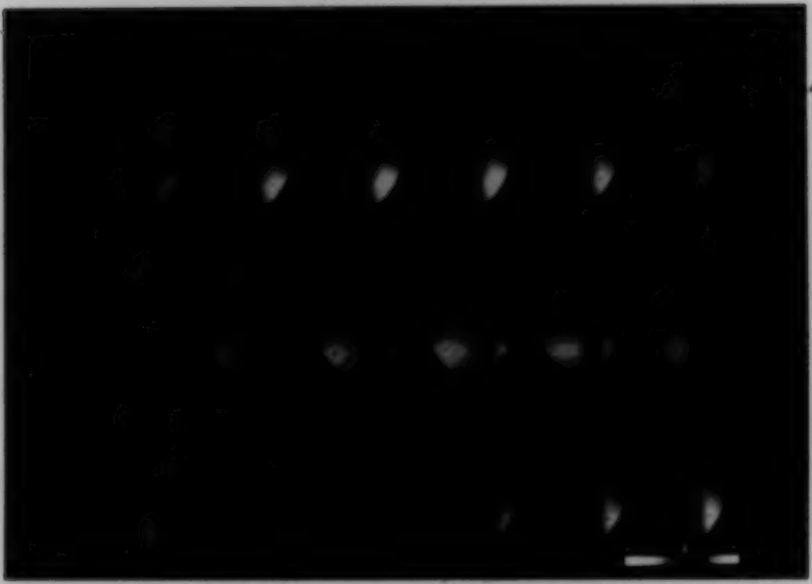
Figure C.6 is the standard page format of display, which presents the user with a selectable section of 18 topographic maps representative of a contiguous section of raw data.

Typical RDTAPE raw EEG data display

Figure C.4



Larger format for cartooning display
Figure C.5



18 contiguous map page type display
Figure C.6